

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра автоматизації та управління в технічних системах**

«До захисту допущено»

Завідувач кафедри

_____ О.І. Ролік

«__» _____ 2019 р.

**Дипломний проект
на здобуття ступеня бакалавра
з напрямку підготовки 6.050201 «Системна інженерія»
на тему: «Автоматизована система керування мікротеплицею»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІА-52

Панасюк Олексій Олегович

Керівник:

Ст. вик. Моргаль О. М.

Рецензент:

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2019 рік

АНОТАЦІЯ

Панасюк О.О. Автоматизована система керування мікротеплицею, КПІ ім. Ігоря Сікорського, Київ, 2019.

Проект містить 60 с. тексту, 38 рисунків, 3 таблиці, посилання на 15 літературних джерел, додатки та 5 конструкторських документів.

Ключові слова: автоматизація, автоматизована теплиця, мікротеплиця, теплиця, гроубокс, штучне освітлення рослин, C, C#.

Об'єктом розробки є схема з мікроконтролером та програмою керування теплицею, додаток для налаштування роботи теплиці.

Мета роботи: створення мікротеплиці, здатної забезпечувати необхідні кліматичні умови для рослин, навіть за умови холодного навколишнього середовища.

Дипломний проект присвячений розробці автоматизованої системи керування мікротеплицею на мові C для мікроконтролера та мові C# для додатку налаштування. Були досліджені основні потреби життєдіяльності рослин та знайдені шляхи їх задоволення. Система автоматизації була успішно протестована під час вирощування екзотичної рослини.

В результаті була створена автоматизована система керування мікротеплицею, якій можна змінювати налаштування завдяки додатку.

SUMMARY

Panasyuk O.O. Automated greenhouse control system, KPI them. Igor Sikorsky, Kyiv, 2019.

The project contains 60 pages, 38 figures, 3 tables, links to 15 literary sources, annexes and 5 design documents.

Key words: automation, automated greenhouse, micro heat, greenhouse, groovox, artificial lighting of plants, C, C #.

The object of the development is the scheme with the microcontroller and the program of management of the greenhouse, an application for adjusting the work of the greenhouse.

The purpose of the work: the creation of a micro-heater, capable of providing the necessary climatic conditions for plants, even in the case of a cold environment.

This work is devoted to the development of an automated control system for the microthermal in C for the microcontroller and the C # language for the configuration application. The main requirements of plant life were investigated and the ways of their satisfaction were found. The automation system has been successfully tested during the cultivation of an exotic plant.

As a result, an automated microcontroller management system was created, where application settings can be changed.

Зміст

ПЕРЕЛІК СКОРОЧЕНЬ	5
ВСТУП	6
Мета, завдання, доцільність розробки об'єкту проектування	6
Методи дослідження, новизна та обґрунтування основних проектних рішень або напрямків досліджень	8
Практичне значення та можливі галузі застосування результатів роботи.	9
1 ОГЛЯД І ПОРІВНЯННЯ ІСНУЮЧИХ РІШЕНЬ	10
1.1 Основні поняття	10
1.2 Огляд та порівняння існуючих рішень	10
1.2.1 Промислова теплиця	11
1.2.2 Мінітеплиця(парник)	12
1.2.3 Мікротеплиця Aerogarden Classic 6-Pod	13
1.2.4 Гроубокс Secret Jardin Hydro Shoot 40x40x120 cm	14
1.2.5 Гроубокс Hortosol 80x80x160 cm	15
1.2.6 Гроубокс ProBox Propagator M, 80x60x40см.....	16
1.2.7 Гроубокс Led 50 з автоматичним крапельним поливом	17
1.3 Висновки	19
2 ПОСТАНОВКА ТА АЛГОРИТМ РОЗВ'ЯЗКУ ЗАДАЧІ.....	20
2.1 Мета дипломної роботи.....	20
2.2 Огляд літератури	20
2.3 Задачі роботи	20
2.4 Цілі роботи, ключові фактори успіху результатів роботи.....	21
2.5 Вихідні дані до роботи	21
2.6 Розгорнуті вимоги до системи.....	21
2.7 Висновки	22
3 ДОСЛІДЖЕННЯ ПОТРЕБ РОСЛИН ТА МЕТОДИ ЇХ ЗАДОВОЛЕННЯ	23

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		2

3.1 Загальний опис	23
3.2 Освітлення.....	23
3.3 Склад повітря.....	27
3.4 Вологість ґрунту.....	28
3.5 Температура.....	29
3.6 Висновки	30
4 АНАЛІЗ ТА РОЗРОБКА СИСТЕМИ.....	31
4.1 Етапи виконання роботи	31
4.2 Основні поняття	31
4.3 Обґрунтування обраних платформ, мов, фреймворків та компонентів	32
4.3.1 Додаток на мові C# для ОС Windows.....	32
4.3.2 Мова C для програми на мікроконтролері	32
4.3.3 Мікроконтролер ATmega328P	32
4.3.4 Польовий транзистор IRLZ44N	33
4.3.5 Біполярний транзистор BC547 C	34
4.3.6 Датчик температури DS18B20	34
4.3.7 Датчик вологості ґрунту Capacitive Soil Moisture Sensor v1.2	35
4.3.8 Датчик температури та вологості повітря DHT22	36
4.3.9 Годинник реального часу DS1302	36
4.3.9 Нагрівальний карбоновий шнур	37
4.3.10 LED стрічка з світлодіодами SMD5054	38
4.3.11 Електромагнітний клапан 12B Atiker.....	39
4.3.12 Повітряний насос 12B MINI PUMP.....	40
4.3.13 LED димер Foton RF DMR 12V 6A	41
4.3.14 Стабілізатор напруги L7805CV	41
4.3.15 Блок живлення 12В 10А	42
4.3.16 USB-UART / USB-TTL перетворювач на мікросхемі CP2102	42
4.3.17 Екструдований пінополістирол	43

4.4 Хід розробки системи	44
4.5 Результати роботи	54
4.5.1 Макет теплиці з макетною платою.....	54
4.5.2 Додаток для ОС Windows.....	56
4.6 Варіант подальшого розвитку роботи.....	57
4.7 Висновки	58
ВИСНОВКИ	59
ПЕРЕЛІК ПОСИЛАНЬ.....	60

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		4

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП – аналогово-цифровий перетворювач

ОС – операційна система

ВСТУП

Мета, завдання, доцільність розробки об'єкту проектування

З давніх часів люди почали вирощувати рослини. Спочатку це було обумовлено необхідністю добування більшої кількості їжі, проте згодом почало поширюватися вирощування декоративних рослин – квітів.

Для нормального росту рослинам необхідні певні умови: достатнє освітлення, вологість ґрунту, певна температура навколишнього середовища, наявність складових повітря у певній відсотковій кількості(кисень, вуглекислий газ). Все це призводить до необхідності догляду за ними, особливо, якщо це екзотичні рослини, які потребують специфічних умов, не притаманних кліматичній зоні, у якій їх хочуть вирощувати.

У нашому бурхливому житті люди постійно чимось зайняті і часу вкрай не вистачає. Також можливі тривалі від'їзди, відрядження, відпустки. Увесь цей час необхідно забезпечувати правильний полив рослин, стежити, щоб вони отримували необхідну кількість світла, особливо взимку, коли світловий день найбільш короткий. Надзвичайно важливо підтримувати необхідну температуру для конкретного виду рослини, адже вона може загинути при занадто низькій або високій температурі. Слід відзначити, що задля підтримання сталої температури необхідно звести стінки довкола рослини, а це створює наступну проблему – відсутність циркуляції повітря та дефіцит або профіцит(в залежності від ситуації) вуглекислого газу. Тож необхідно також забезпечувати рослину повітрям з навколишнього середовища, яке містить необхідну кількість складових у відсотковому відношенні.

Метою даного проекту є вирішення усіх вищезгаданих проблем – створення автоматизованої мікротеплиці, яка буде при мінімальному втручанні людини створювати необхідний мікроклімат для рослин навіть при холодному навколишньому середовищі.

					IA52.200БАК.005 ПЗ	Лист
						6
Ізм.	Лист	№ докум.	Підпис	Дата		

Для вирішення даної проблеми необхідно поставити та виконати певні завдання:

1. Дослідити усі потреби рослин для нормального росту та розвитку
2. Дослідити існуючі рішення та виокремити усі їх переваги та недоліки задля того, щоб зробити висновки та усунути недоліки у розроблюваній системі
3. Розробити загальну архітектуру системи
4. Зробити розрахунки необхідних умов для життя рослин, такі як кількість світла, температурний режим, склад повітря, вологість ґрунту
5. Дослідити наявні електронні компоненти на ринку, здатні забезпечувати необхідні умови, навчитися з ними коректно працювати
6. Розробити принципову електричну схему, перевірити її працездатність та створити макет
7. Зробити розрахунки теплиці та реалізувати її
8. Розробити програмне забезпечення для керування теплицею та її налаштування

					IA52.200БАК.005 ПЗ	Лист
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

Методи дослідження, новизна та обґрунтування основних проектних рішень або напрямків досліджень

Для вирішення даної проблеми були досліджені існуючі рішення, виокремлені недоліки та переваги. Було вирішено об'єднати усі переваги та усунути недоліки задля створення максимально автоматизованої теплиці.

Для підтримання необхідної температури всередині мікротеплиці потрібно, щоб вона була обмежена стінками. З метою забезпечення теплоізоляції було вирішено використати екструдований пінополістирол у якості матеріалу для стінок теплиці. Це дозволило вирощувати рослини навіть при низьких температурах навколишнього середовища.

Закриті стінки, звісно, перешкоджають потраплянню світла від сонця, проте завдяки штучному освітленню тепер відсутня необхідність обов'язкового розташування теплиці під сонцем та вирішена проблема короткого світлового дня взимку.

Використання екструдованого пінополістиролу у якості стінок також дозволило значно зменшити вагу даної теплиці – це дозволяє обслуговувати або переносити її одній особі без допомоги або додаткових інструментів.

Підтримка необхідної температури всередині забезпечується завдяки нагрівальному шнуру, а надходження свіжого повітря, насиченого необхідною кількістю кисню та вуглекислого газу, забезпечує повітряний насос малої потужності, що подає повітря всередину невеликими порціями декілька разів на день. Останнє рішення не впливає серйозно на температуру повітря всередині та дозволяє потроху провітрювати внутрішній простір, не дозволяючи утворюватися грибкам, плісняві та іншим наслідкам замкнених вологих приміщень.

Для керування мікротеплицею був обраний мікроконтролер ATmega328P через його функціональність та кількість ніжок вводу/виводу. У даного мікроконтролеру є порт, що можна налаштувати для виконання

					IA52.200БАК.005 ПЗ	Лист
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

АЦП – це дозволило використовувати у системі аналогові датчики без додаткових зовнішніх компонентів АЦП.

Для налаштування мікротеплицею було обрано створення додатку для ОС Windows – так як це найбільш популярна операційна система.

Новизною даної системи є те, що вона підтримує усі необхідні показники мікрокліматичних умов, у той час, як інші рішення забезпечують лише деякі умови.

**Практичне значення та можливі галузі застосування
результатів роботи**

Створена система може використовуватися для вирощування екзотичних рослин, які потребують специфічних умов, у холодних кліматичних зонах. Також дана теплиця може використовуватися для пророщення розсади на початку року, коли прохолодно та короткий світловий день.

Також дана системи виділяється своєю низькою вартістю та збільшеним функціоналом відносно інших аналогічних рішень.

1 ОГЛЯД І ПОРІВНЯННЯ ІСНУЮЧИХ РІШЕНЬ

1.1 Основні поняття

Теплиця – конструкція, у якій вирощують рослини, які потребують регульованих кліматичних умов. Дані конструкції можуть бути як невеликими так і бути великими промисловими приміщеннями. Зазвичай просто теплицею називають досить великі приміщення на багато рослин.

Теплиці є досить перспективними, адже дозволяють вирощувати рослини, незважаючи на несприятливі кліматичні умови, а також збирати по декілька врожаїв на рік. Також вони є досить дешевими, що дозволило набути їм популярності та розповсюдженості серед багатьох прошарків населення – від дачників і до великих промислових корпорацій.

Мінітеплиця – конструкція для вирощування рослин. Зазвичай невеликих розмірів, розрахована на кількість від декількох до десятків невеликих рослин.

Мікротеплиця – конструкція для вирощування рослин. Розрахована на вирощування однієї конкретної рослини.

Гроубокс – підвид мікротеPLICЬ. Також розподіляється за розмірами на стандартний гроубокс, гроукомпакт та стелс. Зазвичай має закриті стінки. Також досить часто має штучну вентиляцію, освітлення, проте часто відсутній автоматичний полив.

1.2 Огляд та порівняння існуючих рішень

Список теплиць:

1. Промислова теплиця
2. Мінітеплиця(парник)
3. МікротеПлиця Aerogarden Classic 6-Pod
4. Гроубокс Secret Jardin Hydro Shoot 40x40x120 cm
5. Гроубокс Hortosol 80x80x160 cm
6. Гроубокс ProBox Propagator M, 80x60x40cm
7. Гроубокс Led 50 з автоматичним крапельним поливом

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		10

1.2.1 Промислова теплиця



Рисунок 1.1 – сайт teplitsamarket.com.ua

Сайт: <https://teplitsamarket.com.ua>

Опис: Дана теплиця досить великого розміру, що обумовлює певну категорію споживачів – або великі підприємства, або люди, що вирощують усю рослинну їжу для себе на городі. Дана система обладнана захисною плівкою, яка дозволяє зберігати тепло та працювати на манер парника, додатковим освітленням, що дозволяє вирощувати рослини навіть при короткому світловому дні та автоматичним поливом, що дозволяє на значний час залишати теплицю без нагляду.

Переваги: великі розміри теплиці забезпечують великий врожай, додаткове освітлення, полив.

Недоліки: Немає підігріву, великий розмір конструкції влаштує не кожного користувача, необхідна велика площа для встановлення, висока ціна.

Ціна: 45 450 грн.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		11

1.2.2 Мінітеплиця(парник)

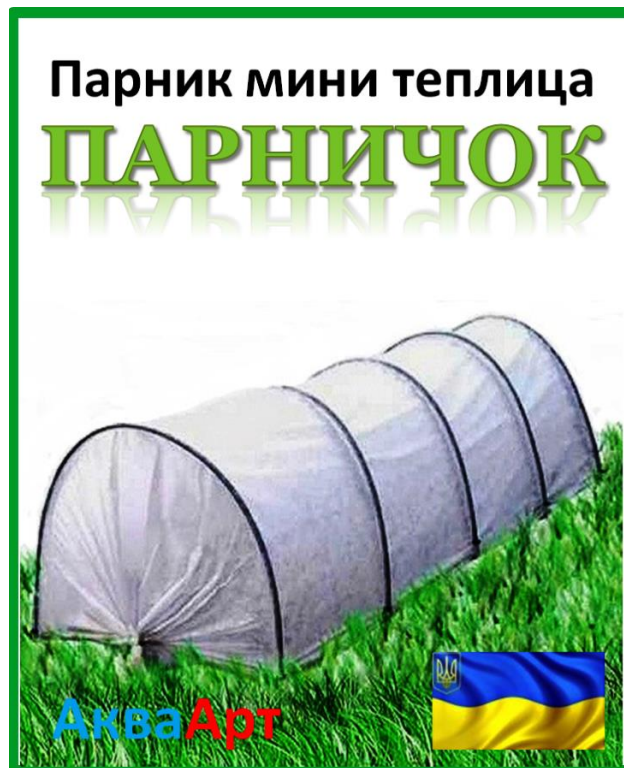


Рисунок 1.2 – Сайт aquaart.in.ua

Сайт: <https://aquaart.in.ua/>

Опис: Дана мінітеплиця є проміжною ланкою між повноцінними великими промисловими теплицями та мікротеплицями для однієї рослини. Вона досить мала, що дозволяє легко її розташувати на прибудинковій ділянці, однак позбавлена автоматизації підтримання мікроклімату - серед атрибутики теплиць наявна лише захисна плівка, яка створює ефект парника – тепло від сонця накопляється всередині.

Переваги: невеликі розміри, мала ціна.

Недоліки: відсутнє додаткове освітлення, полив, підігрів, провітрювання.

Ціна: 525 грн.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		12

1.2.3 Мікротеплиця Aerogarden Classic 6-Pod



Рисунок 1.3 – Сайт megazakaz.com

Сайт: <https://megazakaz.com>

Опис: Даний виріб є прикладом мікротеплиць з відсутніми стінками. Ця риса дозволяю основним джерелом світла залишати сонце, проте унеможлиблює коригування температури для рослини – а отже дані теплиці не придатні для умов з низькими температурами, адже рослина загине. Також дана система не оснащена автоматизацією поливу, що унеможлиблює залишання рослин надовго без нагляду. Серед позитивних рис є те, що присутнє додаткове освітлення, що дозволяє вирощувати рослину навіть при короткому світловому дні.

Переваги: сучасний дизайн, невеликі розміри, освітлення.

Недоліки: немає підігріву, немає поливу, висока ціна для такого малого функціоналу.

Ціна: 4450 грн.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		13

1.2.4 Гроубокс Secret Jardin Hydro Shoot 40x40x120 cm



Рисунок 1.4 – Сайт growmarket.com.ua

Сайт: <https://growmarket.com.ua>

Опис: Дана система є прикладом мікротеплиць, а саме гроубоксів. Вона має закриті стінки, що дозволяє зберігати тепло, проте відсутній підігрів внутрішнього середовища. Також через закриті стінки наявне штучне освітлення, що позбавляє необхідності розташовувати теплицю під сонцем та вирішує проблему короткого світлового дня. Необхідно також відзначити наявність штучної вентиляції, що вирішує проблему із вуглекислим газом та киснем у замкненому просторі.

Переваги: невеликі розміри, освітлення, вентиляція, невелика ціна

Недоліки: немає поливу, підігріву

Ціна: 1650 грн.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		14

1.2.5 Гроубокс Hortosol 80x80x160 см



Рисунок 1.5 – Сайт hydroponics.in.ua

Сайт: <http://hydroponics.in.ua>

Опис: Даний варіант є спрощеним гроубоксом – у нього відсутні полив, вентиляція, освітлення. Задля освітлення спроектовані лише кріплення для світильників. Серед переваг є лише закриті стінки, що дозволяють створити підтримання необхідної температури. Є дуже важливий недолік – висока ціна, незважаючи на повну відсутність автоматизації.

Переваги: невеликі розміри, теплоізоляція, кріплення для освітлення

Недоліки: відсутні освітлення, вентиляція, полив, висока ціна

Ціна: 3 850 грн.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		15

1.2.6 Гроубокс ProBox Propagator M, 80x60x40см



Рисунок 1.6 – Сайт 4grow.com.ua

Сайт: <https://4grow.com.ua>

Опис: Дана теплиця є прикладом невеликого гроубоксу, призначеного для пророщення рослин. Вона має легкий каркас на який можна закріпити штучне освітлення. Також відсутня автоматизована вентиляція, полив, підігрів. Закриті стінки захищають від холодного повітря.

Переваги: легка, невелика конструкція.

Недоліки: відсутнє освітлення, полив, вентиляція, висока ціна.

Ціна: 1 850 грн.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		16

1.2.7 Гроубокс Led 50 з автоматичним крапельним поливом

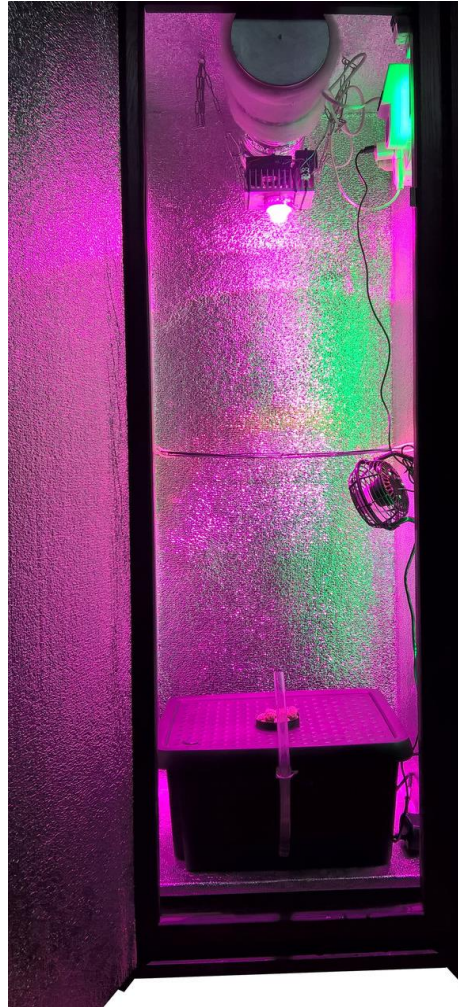


Рисунок 1.7 – Сайт growboxua.com.ua

Сайт: <https://growboxua.com.ua>

Опис: Дана теплиця є найбільш автоматизованою – вона містить штучне освітлення у фіто спектрі, має автоматизований полив та вентиляцію. Також вона має захисні стінки, що дозволяють захистити рослину від холодного повітря. Проте все ще є певні недоліки – відсутній підігрів та зависока ціна.

Переваги: невеликі розміри, освітлення, полив, вентиляція

Недоліки: немає підігріву, зависока ціна

Ціна: 7 100 грн.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		17

В таблиці 1.1 порівнюються перераховані теплиці за функціональністю та характеристиками.

Теплиця	Освіт- лення	Тепло- ізоляція	Піді- грів	Венти- ляція	По- лив	Опти- мальна ціна
Промислова теплиця	+			+	+	
Мініте- плиця (парник)						+
Мікроте- плиця Aerogarden Classic 6- Pod	+			+		
Гроубокс Secret Jardin Hydro Shoot 40x40x120 cm	+	+		+		+
Гроубокс Hortosol 80x80x160 cm		+		+		
Гроубокс ProBox Propagator М, 80x60x40cm		+				
Гроубокс Led 50 з автоматич- ним крапельним поливом	+	+		+	+	

Таблиця 1.1 – Порівняння теплиць

1.3 Висновки

З розглянутих вище прикладів реалізації теплиць можна зробити висновок, що кожна з них має свої переваги та недоліки: у багатьох немає додаткового освітлення та поливу, деякі нехтують теплоізоляцією та вентиляцією, підігріву для підтримання температури взагалі немає ні в кого.

Найбільш широким спектром функціональності володіє Гроубокс Led 50: він має штучне освітлення, теплоізоляцію, вентиляцію для провітрювання та автоматичний полив. Проте дана модель не має підігріву – а отже не може повноцінно підтримувати мікроклімат при холодному навколишньому середовищі. Окрім цього, дана модель не повністю автоматизована, а обладнана механічним таймером, що призводить до того, що користувач не може занадто довго бути відсутнім. До того ж, ціна зависока, незважаючи на досить непоганий функціонал.

Отже, проаналізувавши доступні рішення, було прийнято рішення розробити власний варіант мікротеплиці(гроубоксу) з максимально повним функціоналом, автоматизацією, та невеликою ціною.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		19

2 ПОСТАНОВКА ТА АЛГОРИТМ РОЗВ'ЯЗКУ ЗАДАЧІ

2.1 Мета дипломної роботи

Метою дипломної роботи є створення автоматизованої мікротеплиці з освітленням, підігрівом, поливом для вирощування екзотичних рослин, а також швидкого пророщування розсад.

2.2 Огляд літератури

Основною літературою була документація по мікроконтролеру ATmega328P, датчикам та виконавчим елементам. При роботі з цифровими датчиками необхідно було вивчити протоколи спілкування з ними. Також була використана документація по фреймворкам Atmel Studio, Microsoft Visual Studio.

Також вивчалися статті, що містили інформацію про вирощування рослин та необхідні умови для цього – особливості кліматичних умов: освітлення, вологість землі, температура, . Взяті до уваги також були статті, у яких були описані альтернативні реалізації.

2.3 Задачі роботи

Задачами дипломної роботи є:

1. Дослідження теми.
2. Аналіз наявних рішень.
3. Проектування архітектури системи.
4. Проектування, реалізація та тестування макету.
5. Проектування, реалізація та тестування додатку для ОС Windows.

					IA52.200БАК.005 ПЗ	Лист
						20
Ізм.	Лист	№ докум.	Підпис	Дата		

2.4 Цілі роботи, ключові фактори успіху результатів роботи

Основними цілями дипломної роботи є автоматизація усіх процесів догляду за рослинами задля економії часу та можливості тривалих від'їздів. Ключовими факторами успіху є повна автоматизація теплиці та вирощення тестової рослини з підтриманням усіх необхідних кліматичних умов для перевірки працездатності мікротеплиці.

2.5 Вихідні дані до роботи

Вихідні дані до проекту:

1. Тема для розробки роботи, затверджена кафедрою АУТС.
2. Розробка корпусу макету теплиці.
3. Розробка принципової, структурної функціональної схеми керування мікротеплицею та програми для мікроконтролера, розробка додатку для ОС Windows для налаштування режиму роботи теплиці.
4. Тестування теплиці.

2.6 Розгорнуті вимоги до системи

Зважаючи на увесь можливий функціонал теплиць, слід виокремити такі функції:

1. Штучне освітлення
2. Підігрів
3. Полив
4. Вентиляція

Таким чином, буде створена система, що буде підтримувати увесь перерахований функціонал, буде мати теплоізоляцію та можливість налаштовуватися завдяки додатку на ОС Windows.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		21

Також додатковими вимогами є невелике енергоспоживання та мінімальна ціна.

2.7 Висновки

Сформульовано мету дипломного проекту, основні задачі та цілі. Визначено основні вимоги до системи, яка має бути реалізована.

3 ДОСЛІДЖЕННЯ ПОТРЕБ РОСЛИН ТА МЕТОДИ ЇХ ЗАДОВОЛЕННЯ

3.1 Загальний опис

Рослини є живими організмами, а отже у них є свої потреби для нормально існування. Якщо не дотримувати для них усіх норм – вони загинуть, або будуть вкрай непродуктивно рости. Даний проект вирішує саме ці питання – задоволення всіх ключових потреб рослин задля їх продуктивного росту. Основними факторами, що сприяють нормальному плину життя рослин є:

1. Освітлення
2. Вологість ґрунту
3. Температура
4. Склад повітря

3.2 Освітлення

Майже усі живі організми потребують світла для повноцінного обміну речовин. Рослинам світло потрібне для фотосинтезу – перетворення вуглекислого газу та води на глюкозу. Глюкоза слугує як поживна речовина для рослини та запасасться рослиною, допоки вона знаходиться під світлом. Звісно, не можна постійно тримати рослину під світлом – це її вб'є, адже вона просто виснажиться. Занадто короткий світловий день також погано скажеться на рослині – вона не буде встигати запитися необхідною кількістю поживних речовин. Тож треба дотримуватися певного режиму світлового дня для конкретної рослини задля її нормально росту.

Світло характеризується своєю довжиною хвилі – це є основною характеристикою для нього.

					IA52.200БАК.005 ПЗ	Лист
						23
Ізм.	Лист	№ докум.	Підпис	Дата		

Як відомо, світло розподіляється на такі основні спектральні класи:

1. Ультрафіолетове(нижче 380 нм)
2. Фіолетове(380-430 нм)
3. Синє(430-490 нм)
4. Зелене(490-570 нм)
5. Жовте(570-600 нм)
6. Червоне(600-780 нм)
7. Інфрачервоне(вище 780 нм)

Кожен з цих спектрів має свій власний вплив на рослини і не кожен з них є сприятливим. Розглянемо впливи спектрів на ріст рослин, щоб з'ясувати, які спектри їм необхідні для життя та є безпечними:

1. 280-320 нм – надзвичайно шкідливий для росту рослини
2. 320-400 нм – дуже слабо починається фотосинтез.
3. 400-450 нм – максимально сприяє фотосинтезу
4. 490-565 нм – не має особливого впливу на рослини
5. 565-640 нм – слабкий фотосинтез
6. 640-700 нм – максимальний фотосинтез
7. 700-780 нм – подовжує цвітіння та подовжує стебло, проте не впливає на фотосинтез

Отже, виходячи з цих даних можемо зробити висновок, що для повноцінного росту рослин необхідний такий спектр світла – 430–480 та 650–680 нм. Освітлюючи рослину лише даним діапазоном світла можна максимально ефективно використовувати енергоресурси, не завдаючи шкоди рослині.

Окрім спектру світла є також така характеристика як сила світлового потоку. Вона вимірюється у Люменах. Також розрізняють густину світлового потоку. Вона вимірюється у Люксах. 1 Люкс дорівнює освітленості поверхні у 1 м² при світловому потоці 1 Люмен. У різних видів рослин різні потреби у світлі. Отже, наведемо приклади різноманітних рослин з різними потребами у світлі.

					IA52.200БАК.005 ПЗ	Лист
						24
Ізм.	Лист	№ докум.	Підпис	Дата		

У таблиці 3.1 відображені різні роди кімнатних рослин за декількома категоріями потреби у світлі:

Рекомендована освітленість, лк	Рослини
2500-3000	Агавові, Акантові, Аралієві, Виноградні, Губоквітні, Коммелінові, Марантові, Молочайні, Папороті, Тутові
3000-4000	Аїзонові, Бегонії, Вербенові, Камнеломкові, Маренові, Норичникові, Пальмові, Пасленові, Паречні, Саговникові, Чайні, Епіфітні кактуси
4000-6000	Амарілісові, Бананові, Бігтонієві, Бобові, Верескові, Гранатові, Ластовневі, Мальвові, Орхідні, Пеларгонієві, Складноцвітні, Стеркулієві, Толстянкові
Понад 6000	Кактуси, Кутрові, Маслинні, Миртові, Нічноцвітні, Рутові, Страстоцвітні

Таблиця 3.1 – Інформація про рослини та їх потреби зі статті <https://лампочка.com/articles/skolko-sveta-neobkhodimo-rasteniyam/>

Як бачимо, рослинам необхідний світловий потік у 2-6 тис. Люксів, тобто 2-6 тис. Люменів на метр квадратний.

Побічним результатом фотосинтезу, який відбувається під дією світла, є вироблення кисню. Це, звісно, змінює склад повітря у навколишньому середовищі. Докладніше про повітря у наступному підрозділі.

3.3 Склад повітря

Живі організми поділяються на аеробні та анаеробні. Анаеробні організми не потребують кисню для життєдіяльності, в той час як для аеробних організмів кисень є одним з головних та необхідних хімічних елементів для життєдіяльності.

Повітря складається з багатьох компонентів:

1. Азот(78.08%)
2. Кисень(20.95%)
3. Аргон(0.93%)
4. Вуглекислий газ(0.02-0.04%)
5. Багато інших

Серед цих газів найбільш важливими для рослин є кисень та вуглекислий газ. Вуглекислий газ разом із водою бере участь у виробництві глюкози з побічним утворенням кисню. Тобто вуглекислий газ необхідний для запасання рослиною поживними речовинами.

Проте не варто вважати, що кисень це лише побічний продукт. До 50% кисню, який рослина виробила впродовж світлового дня, вона споживає вночі, дихаючи.

Варто також зазначити, що для рослин важливий склад повітря не лише за його складовими, але й за їх відсотковим відношенням. Відсоткове відношення складових повітря, яке врівноважалося в наші дні є ідеальним для рослин, адже процес врівноваження тривав мільйони років і рослини постійно адаптувалися під нього.

Отже, було з'ясовано, що для нормальної життєдіяльності рослин необхідно підтримувати певне відсоткове відношення складових повітря.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		27

3.4 Вологість ґрунту

Вода необхідна усім живим організмам – завдяки ній здійснюється обмін поживних речовин, охолодження організму через випарювання надлишку води та відбуваються різноманітні хімічні реакції. Як відомо, вода приймає участь у фотосинтезі, тож є надзвичайно важливою для рослин.

Рослини живляться водою завдяки кореневій системі. Існує два види росту корневих систем: ті, що ростуть вглиб, та ті, що ростуть вшир. Перший вид характерний для систематично погано зволоженої землі. Так як води на поверхні немає у такій місцевості, то рослини намагаються кореневищами дістатися до шарів ґрунту, де є підземні води. Другий вид притаманний добре зволоженій землі: вода є близько до поверхні, а отже немає необхідності рости вглиб.

Враховуючи вищесказане, виходить, що для рослин надзвичайно важливо дотримувати певний водний режим ґрунту, у якому вони ростуть.

Для кожної рослини існують свої певні потреби у воді та вологості ґрунту. Дані потреби суто індивідуальні і залежать від кліматичної зони, де росте рослина, пори року так багато іншого.

Загалом можна сказати, що для кожної рослини необхідні свої певні умови і не правильно визначити одну ідеальну вологість для усіх видів – слід підбирати потрібну вологість ґрунту для кожного виду окремо.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		28

3.5 Температура

Температура – надзвичайно важливий показник, як для живих організмів загалом, так і для рослин конкретно. По-перше, при температурі нижче 0 градусів Цельсію вода замерзає, а як відомо, рослини мають значну частку води. Отже, при температурі нижчу 0 градусів Цельсію, клітини рослини руйнуються, рослина гине.

При низьких температурах (від 0 до 10 градусів Цельсію) обмін речовин у рослин настільки сповільнюється, що рослині дуже важко доводиться підтримувати себе у живому стані.

Між 10 та 15 градусами рослини вже можуть нормально рости, проте постійно відчують так званий «стрес». Занадто довге перебування у такому стані може призвести до гибелі рослини.

Між 15 та 20 градусами рослини почувають себе більш менш нормально та повноцінно ростуть, проте для деяких теплолюбивих рослин дана температура може бути прохолодною.

Більшість рослин почувають себе комфортно при температурі від 20 до 30 градусів Цельсію. Саме таку температуру рекомендують підтримувати тим, хто доглядає за рослинами.

При температурі вище 30 градусів рослина також гине через перегрівання – виживають лише незначна кількість видів тропічних рослин, що пристосовані до даних температурних умов в зв'язку зі своїм походження.

Отже, для рослин необхідні спеціальні температурні умови. Хоча, від 20 до 30 градусів є найбільш оптимальною температурою, проте все ж варто враховувати особливості кожного виду індивідуально та створювати найбільш сприятливі умови існування для рослин.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		29

3.6 Висновки

Було з'ясовано, що для всіх видів рослин необхідні свої спеціальні умови. Були вивчені основні потреби рослин, а саме: потреба в освітленні, певному складовому та відсотковому складі повітря, вологості ґрунту та температурі. Також були вивчені загальні відомості про кожну з даних потреб та визначені основні показники, які необхідно забезпечити у теплиці, що буде проектуватися.

Для задоволення потреб в освітленості необхідно створити світловий потік з густиною 2-6 тисяч люменів на метр квадратний. При чому дане світло має містити такий спектр: 430–480 та 650–680 нм. Він забезпечить рослинам нормальний ріст, при незначному енергоспоживанні, адже виключаються всі інші непотрібні спектральні домішки.

При проектуванні замкненої теплиці слід враховувати, що рослина буде сама змінювати відсоткове відношення складу повітря. По елементне коригування складу повітря занадто важке, отже, простіше за все, за допомогою повітряного насоса наганяти повітря з атмосфери всередину теплиці.

Слід також відзначити, що рослини потребують певної температури. Було з'ясовано, що необхідно підтримувати повітря всередині теплиці на рівні 20-30 градусів, в залежності від температури, а отже, обов'язково необхідний нагрівальний елемент.

					ІА52.200БАК.005 ПЗ	Лист
						30
Ізм.	Лист	№ докум.	Підпис	Дата		

4 АНАЛІЗ ТА РОЗРОБКА СИСТЕМИ

4.1 Етапи виконання роботи

Розробка системи була виконана у такі етапи:

1. Постановка задачі
2. Аналіз системи
 - 1) Сбір інформації
 - 2) Аналіз існуючих рішень
 - 3) Обрання необхідного функціоналу для системи
3. Проектування системи
 - 1) Створення архітектури системи
 - 2) Проектування корпусу теплиці
 - 3) Проектування принципової схеми
4. Реалізація системи
 - 1) Побудова корпусу теплиці
 - 2) Пайка макетної плати
 - 3) Тестування макетної плати
 - 4) Створення програми для мікроконтролеру
 - 5) Тестування програми для мікроконтролеру
 - 6) Створення додатку
 - 7) Тестування додатку
 - 8) Тестування готової системи

4.2 Основні поняття

UART – універсальний асинхронний приймач/передавач.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		31

4.3 Обґрунтування обраних платформ, мов, фреймворків та компонентів

4.3.1 Додаток на мові C# для ОС Windows

Операційна система Windows є найбільш популярною у світі. Саме тому вона була обрана для створення додатку для налаштування. C# є досить простою мовою для швидкого створення програмного забезпечення із графічним дизайном так багатопоточністю.

4.3.2 Мова C для програми на мікроконтролері

Зазвичай мікроконтролери програмують такими мовами:

1. Асемблер
2. C
3. C++

Асемблер є досить важкою мовою для реалізації великих алгоритмів.

C++ дуже не оптимальний у використанні пам'яті, а при написанні програм для мікроконтролеру – оптимальне використання пам'яті є однією з найважливіших завдань.

Мова C є одночасно і оптимальною у використанні пам'яті, і досить простою та зрозумілою синтаксично. Саме тому дану мову було обрано для написання програми керування мікроконтролером.

4.3.3 Мікроконтролер ATmega328P

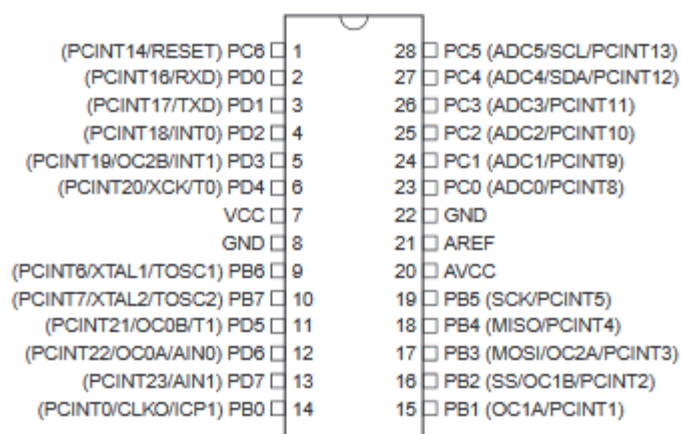


Рисунок 4.1 – Сайт www.alldatasheet.com

Мікроконтролер ATmega328P працює на частоті 8 МГц – цього достатньо для забезпечення нормальної роботи з датчиками, АЦП, та UART.

Також даний мікроконтролер може конфігурувати свій порт C на внутрішню обробку аналогового сигналу, що дозволяє працювати з аналоговими датчиками.

Через велику кількість датчиків та виконавчих елементів виникає необхідність у достатній кількості пінів, що можуть програмуватися на ввід/вивід. Даний мікроконтролер повністю задовольняє дану потребу для підключення необхідних компонентів.

Також даний мікроконтролер має досить невелику ціну, є відомим та його легко знайти у продажі, що є дуже важливим для комерційного продукту.

4.3.4 Польовий транзистор IRLZ44N

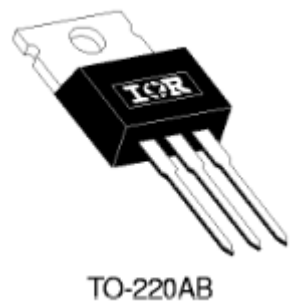


Рисунок 4.2 – Сайт www.alldatasheet.com

Для керування подачею живлення необхідний спеціальний компонент. Даним компонентом було обрано польовий транзистор IRLZ44N. Він позбавлений рухомих частин, що робить його досить надійним при роботі.

Даний транзистор поміщений у досить популярний корпус TO-220AB, який має розсіювальну потужність 83Вт. При тому, що на платі 5-12В, а ток стоку 41А – ми можемо керувати ним досить великою потужністю.

Максимальна напруга стоку-витоку 55 В, а максимальна напруга затвору – 16В, що повністю задовольняє нашим потребам у 12 В.

Завдяки даному польовому транзистору ми керуємо подачею напруги як на малопотужні датчики, так і на багатопотужні виконавчі елементи.

4.3.5 Біполярний транзистор BC547 C



Рисунок 4.3 – Сайт www.alldatasheet.com

Даний транзистор є N-канальним біполярним транзистором.

Він повністю задовольняє наші потреби – завдяки ньому був створений простий датчик наявності води у піддоні.

4.3.6 Датчик температури DS18B20



Рисунок 4.4 – Сайт mini-tech.com.ua

Даний датчик має захисний корпус, що захищає його вод вологи та фізичних ушкоджень.

Температурний діапазон роботи від -55 до +125 градусів Цельсію, чого більш ніж достатньо для наших потреб, адже для рослин необхідно, зазвичай, від 15 до 30-35 градусів Цельсію.

Обмін даних відбувається за протоколом 1-Wire, напруга живлення від 3 до 5.5 В, а струм живлення всього 1мА.

Як бачимо, даний датчик повністю задовольняє наші потреби, малопотужним, а також простим у роботі.

4.3.7 Датчик вологості ґрунту Capacitive Soil Moisture Sensor v1.2



Рисунок 4.5 – Сайт <http://ilogix.in.ua>

Для того, щоб дізнатися, що необхідно здійснити полив – необхідний датчик вологості землі. Існує декілька таких датчиків, однак більшість з них піддаються корозії, через що їх покази ненадійні та через швидкий час вони виходять з ладу.

Даний датчик є ємнісним антикорозійним датчиком – він стабільний у своїх показаннях та довго може працювати.

У даного датчика аналоговий виходить 3 дроти: GND, VCC(+5V) та аналоговий вихід. Саме через даний аналоговий вихід мікроконтролер і зчитує значення та оцифровує їх.

Напруга живлення даного датчику від 3.3 В до 5.5 В, що абсолютно нам підходить. Також цей датчик вологості землі досить мало споживає току, що є дуже важливим при проектуванні подібних систем.

Варто відзначити, що його зависока, як для датчика, проте це повністю компенсується його антикорозійністю.

4.3.8 Датчик температури та вологості повітря DHT22

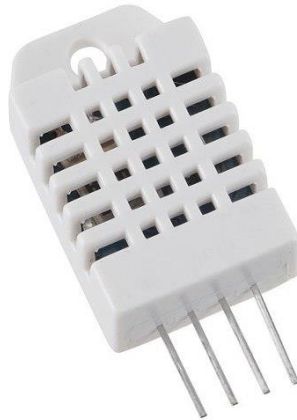


Рисунок 4.6 – Сайт arduino.ua

Даний датчик є цифровим датчиком температури та вологості повітря. DHT22 відноситься до класу вимірювачів підвищеної точності.

Температурний діапазон даного датчику від -40 до 80 градусів Цельсію, що повністю задовольняє нашим потребам. Точність вимірювання температури ± 0.5 градуса Цельсію.

Діапазон вимірювання вологості повітря від 0 до 100 відсотків. Точність вимірювання вологості $\pm 2\%$.

Напруга живлення від 3.6 до 6 В, що задовольняє живленню схеми.

Відноситься до класу ультранизького енергоспоживання, що являється надзвичайно важливим при проектуванні систем.

4.3.9 Годинник реального часу DS1302



Рисунок 4.7 – Сайт mini-tech.com.ua

Даний годинник відзначається своєю дешевизною, порівняно з іншими аналогами.

Пристрій видає такі дані: секунди, хвилини, години, дні, місяці, роки. Є коригування на високосні роки.

4.3.9 Нагрівальний карбоновий шнур

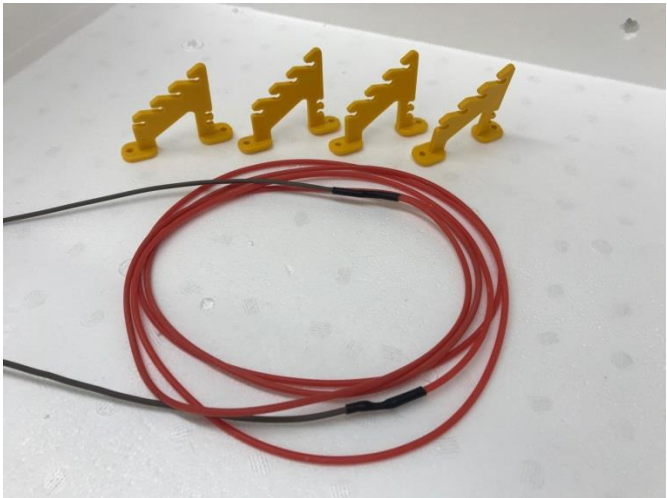


Рисунок 4.8 – Сайт gospshop.com

Для рослин надзвичайно важливо забезпечення необхідних кліматичних умов. Температурний режим є надзвичайно важливим фактором життя рослин. Зазвичай рослини потребують від 15 до 30-35 градусів Цельсію. Даний нагрівальний шнур допомагає підтримувати необхідний рівень температури у теплиці.

Напруга живлення 12 В, струм живлення 4 А, потужність 48 Вт. Цієї потужності достатньо для задоволення потреб рослин у температурному режимі.

Довжина шнуру 210 см, що дозволяє як завгодно зручно розташувати його всередині теплиці.

Шнур ззовні має силіконову оболонку, яка відіграє роль ізолятора, а отже даний нагрівальний шнур захищений від вологи.

4.3.10 LED стрічка з світлодіодами SMD5054



Рисунок 4.9 – Сайт dhgate.com

Важливим фактором для нормально росту рослин є освітлення. Слабке освітлення не буде давати стрімко розвиватися, а занадто інтенсивне освітлення може викликати опіки, або інші негативні наслідки та рослина загине. Необхідно чітко знати яка інтенсивність світла необхідна певній рослині та створювати для неї відповідні умови.

Дана світлодіодна стрічка потребує напругу 12 В та 1.25 А на метр. Тобто її потужність 15 Вт на метр. Яскравість даної стрічки 3000 люменів на метр.

У цій світлодіодній стрічці використовуються світлодіоди SMD5054. Дані світлодіоди світять у 2 рази яскравіше, а також споживають у 2 рази менше енергії ніж відомі яскраві світлодіоди SMD5050.

Використовується 2 метри даної стрічки задля досягнення необхідної яскравості.

Методом експериментів було з'ясовано, що стрічка перегрівається у такому замкненому і погано провітрюваному середовищі, як мікротеpliers, при максимальній яскравості. Експериментально було з'ясовано, що стрічка перестає нагріватися і її температура не перевищує 44 градусів Цельсію при 20% потужності.

Отже, після обмеження по споживанню струму отримаємо такі значення: $1,25 \text{ А} * 2 \text{ метри} * 0,2 = 0,5 \text{ А}$ – це струм, що споживає 2 метри даної світлодіодної стрічки.

Отже, дана стрічка з обмеженнями має потужність $12 \text{ В} * 0,5 \text{ А} = 6 \text{ Вт}$, що є досить мало для повноцінного освітлення.

З урахуванням обмежень, можемо порахувати, що стрічка стала світити з яскравістю 600 люмен на метр – отже 1200 люмен на 2 метри.

4.3.11 Електромагнітний клапан 12В Atiker



Рисунок 4.10 – Сайт lpgtech.ua

Усім живих організмів необхідна вода. При розробці автоматизованих систем зазвичай не використовують водяні насоси, а використовують електромагнітні клапани, вказуючи, що у систему повинна подаватися вода під певним тиском.

Даний електромагнітний клапан, при подачі на нього струму, відкривається та пропускає воду, яка поливає рослини. При відсутності струму клапан автоматично закривається. Це забезпечує безпеку при випадковій пропажі струму у системі.

Напруга для роботи даного клапану 12 В.

Температура роботи від -20 до 110 градусів Цельсія, що задовольняє умовам роботи даної системи.

Даний клапан повністю задовольняє потребам та здатен керовано подавати воду до рослин.

4.3.12 Повітряний насос 12V MINI PUMP



Рисунок 4.11 – Сайт aliexpress.com

Для рослин надзвичайно важливо, щоб у повітрі був і вуглекислий газ, і кисень, причому у достатній кількості, але не занадто багато. Атмосфера нашої планети їм ідеально підходить. Створювати штучне по елементне регулювання усіх показників повітря дуже складно – особливо для даного проекту. Це просто невиправдано завищить ціну кінцевого виробу. Саме тому було вирішено використовувати повітряний насос задля того, щоб наганяти повітря з зовнішнього середовища всередину теплиці.

Необхідна напруга для роботи даного насосу 12 В, струм 0.36А. Потужність 4.3 Вт, що є досить небагато.

Даний насос накачує повітря зі швидкістю 2 літри за хвилину, а отже 120 літрів за годину. Враховуючи, що внутрішній об'єм теплиці 90 літрів – то виходить, що за годину робота даний насос здатен повністю оновити повітря всередині.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		40

4.3.13 LED димер Foton RF DMR 12V 6A

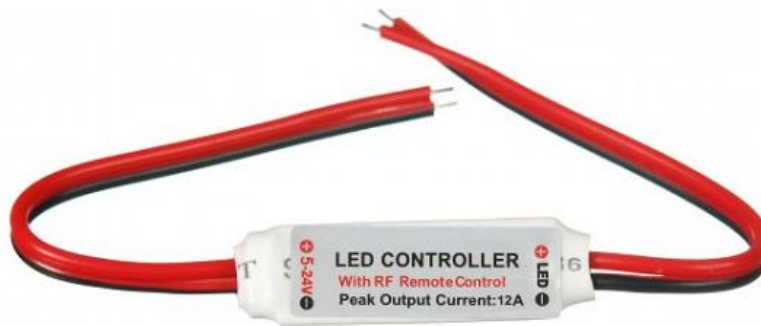
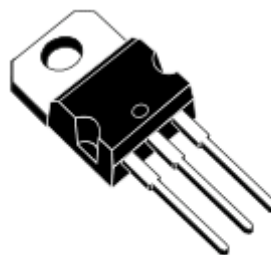


Рисунок 4.12 – Сайт rozetka.com.ua

Даний пристрій необхідний для обмеження струму, що йде на світлодіодну стрічку до 20%. Це треба тому що світлодіодна стрічка починає перегріватися при збільшеній потужності.

Даний димер розрахований на напругу 12 В та максимальний струм 6А. Цього абсолютно достатньо, адже стрічка завдовжки 2 метри споживає 0,25 А при потужності 20%.

4.3.14 Стабілізатор напруги L7805CV



TO-220

Рисунок 4.13 – Сайт www.alldatasheet.com

Даний компонент являє собою стабілізатор напруги.

Максимальна вхідна напруга 35 В. Максимальний струм навантаження 1.5 А. Вихідна напруга 5В.

Діапазон робочих температур від 0 до 125 градусів Цельсію.

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		41

Даного стабілізатору достатньо для забезпечення живленням мікроконтролера та усіх датчиків, адже вони споживають вкрай мало струму.

4.3.15 Блок живлення 12В 10А



Рисунок 4.14 – Сайт led-story.com.ua

Для живлення даної системи нам необхідний блок живлення на 12 В.
Загалом, усі датчики та виконавчі елементи максимально будуть споживати 7 А(при одночасній роботі усіх виконавчих елементів).
Отже, беремо з невеликим запасом по потужності та обираємо даний блок живлення на 10 А.

4.3.16 USB-UART / USB-TTL перетворювач на мікросхемі CP2102



Рисунок 4.15 – Сайт mini-tech.com.ua

Даний перетворювач слугує для того, щоб підключати пристрої, які мають інтерфейс UART до комп'ютерів з інтерфейсом USB.

Для даного пристрою комп'ютер швидко автоматично знаходить драйвер, чого не можна сказати про інші моделі, з якими часто бувають складнощі.

4.3.17 Екструдований пінополістирол



Рисунок 4.16 – Сайт 27.ua

Даний матеріал досить легкий та забезпечує теплоізоляцію для теплиці. Також він захищає від вітру.

4.4 Хід розробки системи

Перш за все були вивчені усі потреби рослин та методи їх заохочення. Був обраний спектр кліматичних параметрів, яку буде підтримувати мікротеplitsя. Після цього було проектування теплиці. Теплиця повинна була відповідати таким конструкторським та технічним вимогам:

1. Нагрівальний елемент мав розташовуватися знизу, щоб гаряче повітря піднімалося нагору
2. Один датчик температури мав розташовуватися внизу, біля нагрівального елемента, щоб не допускати перегрівання останнього, а інший вгорі, щоб отримувати повну інформацію по розподіленню гарячого повітря всередині теплиці.
3. Повітряний насос мав направляти повітря ззовні всередину у верхній частині теплиці, щоб прохолодне повітря зверху опускалося вниз і відбувалася циркуляція повітря.
4. Для забезпечення контрольованого поливу необхідно було розташувати 2 датчики вологості – один датчик вологості ґрунту, щоб знати, коли землю необхідно полити, а другий – датчик наявності води в піддоні, щоб знати, коли необхідно зупинити полив.
5. Потрібно було розрахувати необхідну кількість світла для рослин та забезпечити їх світильником потрібної потужності.
6. Необхідно розмежувати відділення з електронікою та вологе відділення з рослиною задля техніки безпеки

Після визначення переліку необхідних конструкторських рішень було вирішено спершу побудувати макет самої теплиці, а вже потім реалізовувати електронну частину, щоб одразу її протестувати.

					IA52.200БАК.005 ПЗ	Лист
						44
Ізм.	Лист	№ докум.	Підпис	Дата		

Перш за все, було вирішено, що у теплиці буде два отвори, через які можна буде діставати рослину та здійснювати інші дії відносно неї.

Один отвір буде знаходитись спереду, за з'ємною кришкою, а другий зверху, прикритий другою кришкою.

Так як іноді буде кортіти подивитись на рослину, а випускати тепле повітря небажано, то вирішено було, що за передньою кришкою буде знаходитись органічне скло, яке буде вставлятися через верхній отвір і ходити по направляючим. Таким чином, знявши передню кришку, можна подивитися на рослину, не завдавши їй шкоди.



Рисунок 4.17 – Теплиця з передньою закритою кришкою



Рисунок 4.18 – Теплиця зі знятою передньою кришкою та оргсклом, що не випускає тепле повітря

Після цього вирішено було розробити спеціальну кришку, яка б містила в собі відділення для електроніки – цим кроком буде вирішена проблема з технікою безпеки, а саме – електроніка буде відділена від вологого середовища теплиці.

Була розроблена кришка з відділенням, яке мало власну кришку.



Рисунок 4.19 – Верхня кришка теплиці у закритому стані



Рисунок 4.20 – Верхня кришка, знята з теплиці



Рисунок 4.21 – Внутрішнє відділення для електроніки у верхній кришці теплиці.

Було вирішено світлодіодну стрічку кріпити знизу верхньої кришки. Дроти до іншої електроніки було виведено назовні, де вони штекерами з'єднувалися з дротами з верхньої кришки. Це дозволить від'єднувати їх на час знімання кришки. Подібне рішення було застосоване також і для шлангів для повітря та води.

Після розробки корпусу теплиці було вирішено обрати давачі та виконавчі елементи, що повністю задовольнять наші потреби у створення мікроклімату для рослин.

Після вивчення їх характеристик було вирішено, що нагрівальним елементом буде нагрівальний шнур із потужністю 48 Вт. Цього більш ніж повинно вистачити для обігріву та підтримання температури такого малого приміщення. Давачем температури, що буде контролювати, щоб не перегрівся нагрівальний елемент було обрано DS18B20. Його характеристики повністю підходили, до того ж він у захисному корпусі, що захистить його від вологи. Зверху теплиці був закріплений давач DHT22. Його було обрано, адже він доволіно відомий, простий у використанні, а також зчитує одразу два показники – температуру та вологість повітря.

У якості давача вологості землі був обраний давач Capacitance Soil Moisture Sensor v1.2. Головним критерієм відбору була його антикорозійність і, як наслідок, стабільність роботи. Давач наявності води у піддоні вирішено було реалізувати за допомогою двох контактів – на один подається струм, а з іншого зчитуємо логічний нуль або одиницю, щоб дізнатися чи є там вода. Тут була використана властивість води до проведення струму, однак для уможливлення даної системи необхідний був біполярний транзистор, за допомогою якого ми би підсилювали вхід з нашого імпровізованого датчика. Обраного електромагнітного клапану Atiker цілком вистачає для розроблюваної невеликої теплиці.

В решті решт постало питання щодо освітлення. Була обрана світлодіодна стрічка зі світло діодами SMD5054, адже вони є надзвичайно ефективними та мало споживають енергії. Дана стрічка світить у фіто спектрі, тож вона повністю відповідає нашим потребам, щодо освітлення у спектрі 430-480 та 650-680 нм. Однак з'явилася проблема у її використанні. Справа у тому, що якщо подавати на неї струм без обмежень(на 100%), то вона занадто сильно грілася у замкненому, погано провітрюваному приміщенні(теplicі). Експериментальним шляхом було з'ясовано, що

необхідно обмежувати струм до 20%, щоб температура стрічки не підіймалася вище 44 градусів Цельсію. Тож тепер необхідно було розрахувати, скільки метрів стрічки необхідно встановити, щоб було достатньо світла для рослин.

Перш за все необхідно врахувати властивості світло діодів: їх кут свічення 120 градусів, а отже світло від світло діодів буде поглинатися частково стінками теплиці.

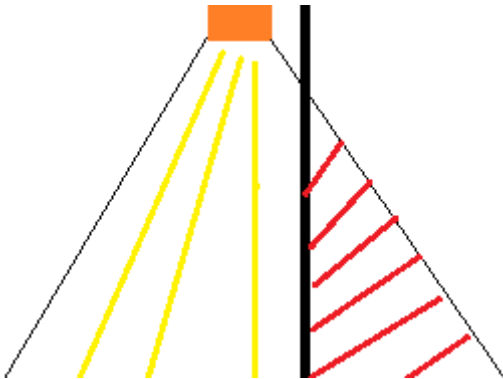


Рисунок 4.22 – Частина світла від світло діода не потрапляє на рослину, а поглинається стінкою

Враховуючи майбутнє розташування світло діодів, а також їх кут свічення 120 градусів – то виходить, що приблизно половина світла буде поглинатися стінками.

Отже, у нас було лише 20% від максимальної потужності світлодіодної стрічки, а після врахування кута свічення отримаємо 10%.

Ми знаємо, що рослинам потрібно 2-6 тис. люксів, тобто 2-6 тис. люмен на метр квадратний, проте площа нашої теплиці 0.09 метра квадратних, тобто приблизно у 10 разів менше.

Іншими словами нам необхідно, щоб світлодіодна стрічка світила з яскравістю приблизно 400-600 люмен. Один метр світлодіодної стрічки світить з яскравістю 3000 люмен при 100%, проте ми зясували, що в результаті отримаємо лише 10% світла. Отже 1 метр стрічки дає 300 люмен. Виходячи з цього було вирішено встановити дану стрічку завдовжки 2 метри.



Рисунок 4.23 – Світлодіодна стрічка, приклеєна до кришки теплиці.

Обравши та протестувавши всі компоненти слід було розробляти плату керування мікротеплицею. Була розроблена структурна, принципова, функціональна схеми а також складальний кресленик та друкована плата.

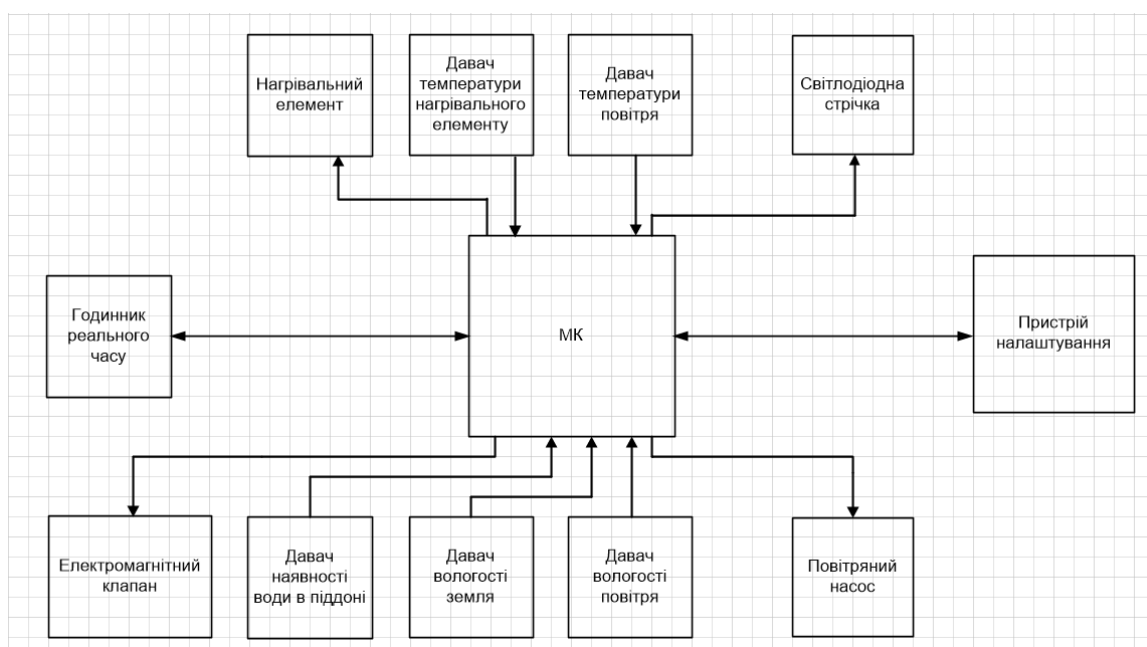


Рисунок 4.24 – Структурна схема

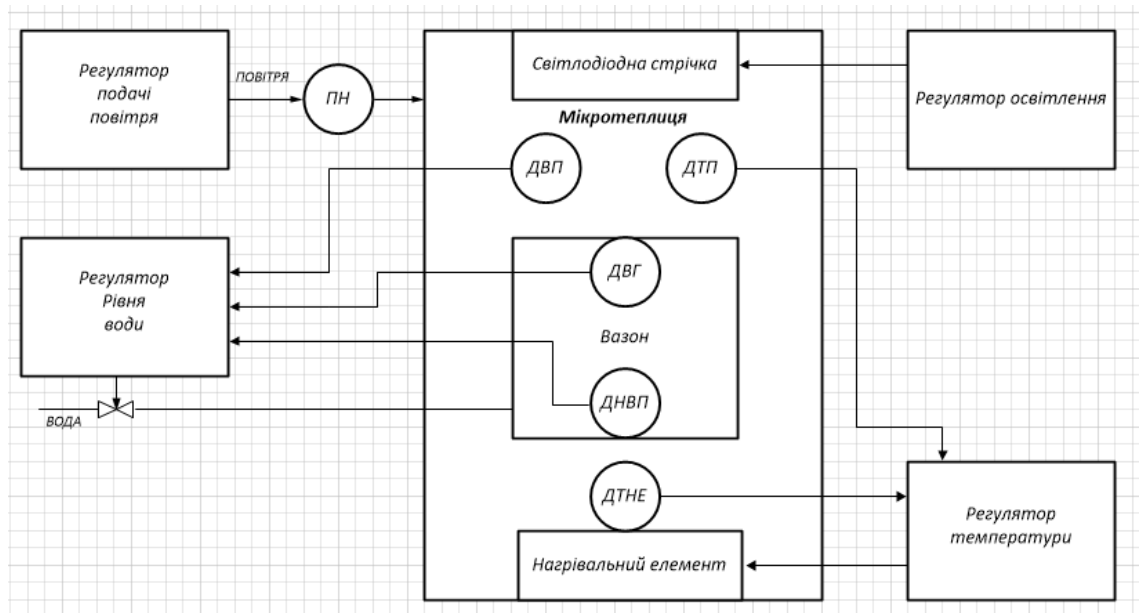


Рисунок 4.25 – Функціональна схема

ПН - повітряний насос

ДВП – давач вологості повітря

ДВГ – давач вологості ґрунту

ДНВП – давач наявності води у піддоні

ДТП – давач температури повітря

ДТНЕ – давач температури нагрівального елемента

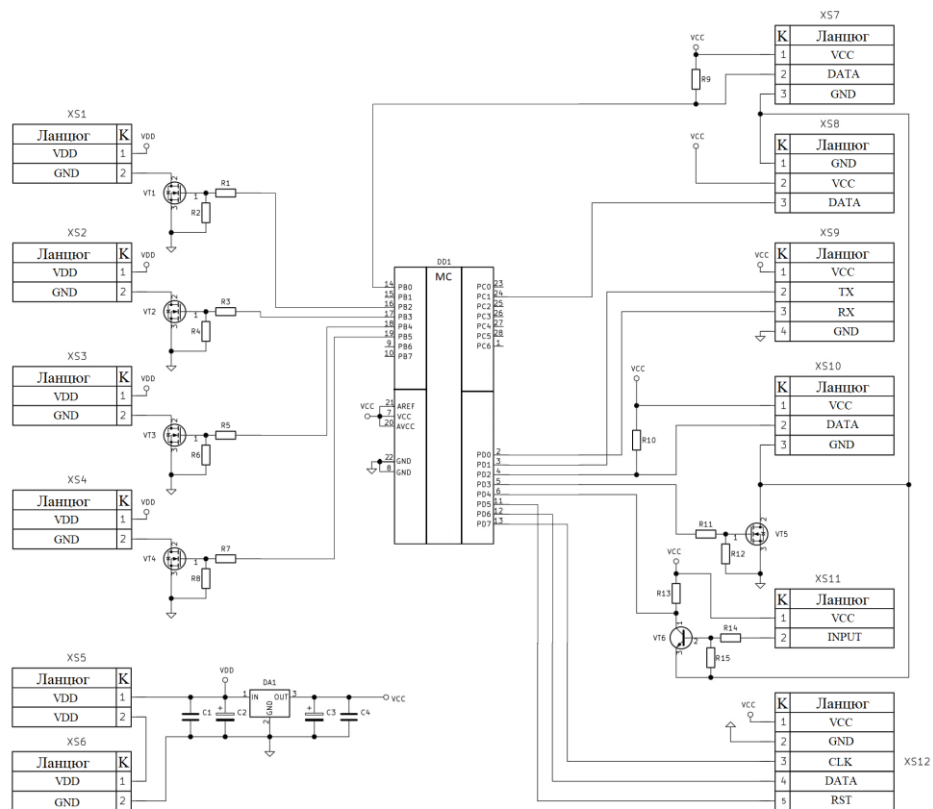


Рисунок 4.26 – Принципова схема

Після розробки схем була побудована 3-д модель майбутньої плати.

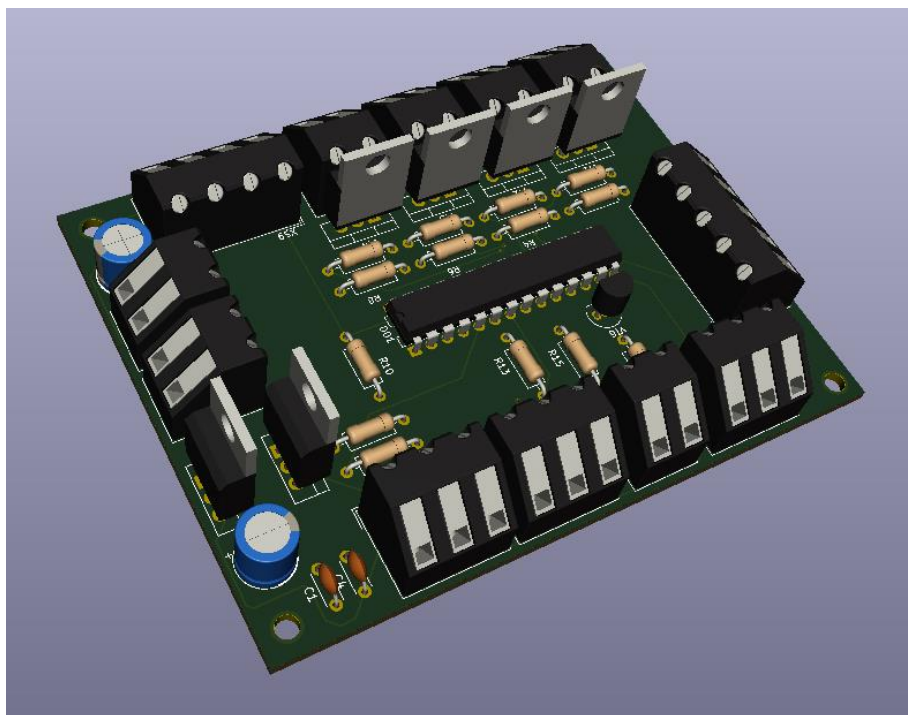


Рисунок 4.27 – 3-д модель плати керування теплицею

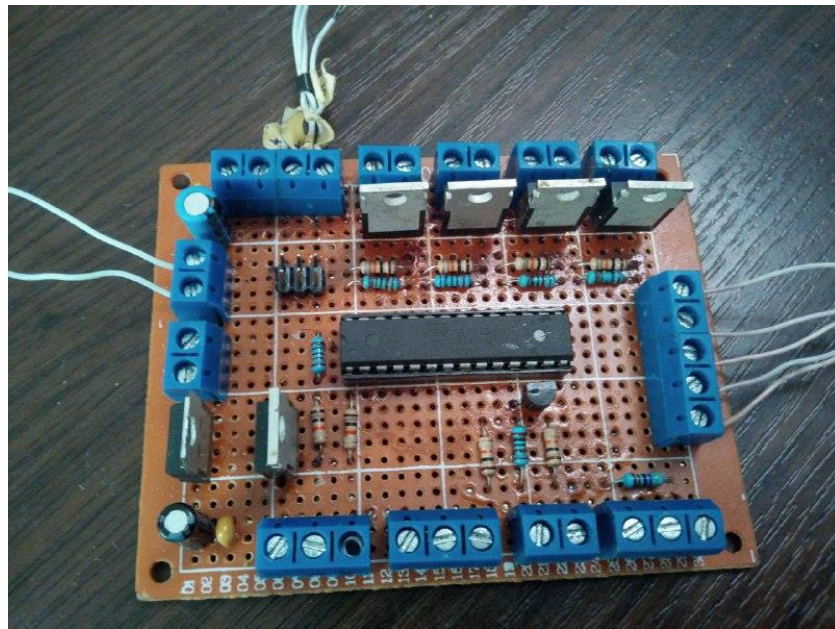
Ізм.	Лист	№ докум.	Підпис	Дата

ІА52.200БАК.005 ПЗ

Лист

52

Після побудови 3-д моделі був спаяний реальний макет для тестування працездатності схеми.



Рисункок 4.28 – Макетна плата

Після цього були розроблені та протестовані програми для мікроконтролера та ОС Windows.

4.5 Результати роботи

У результаті роботи були створені принципова, структурна, функціональна схеми, складальний кресленик, 3-д модель плати, макетна плата, програма для мікроконтролеру, а також додаток для ОС Windows для налаштування роботи теплиці.

Для перевірки працездатності теплиці була посаджена пророщена цибулина тропічної квітки *Scadoxus multiflorus*. Висота рослини при висадці – 8 см. Після висадки заявилися ще два ростки – зліва та справа, проте вони спочатку повільно розвивалися. На 8 день вони почали помітно рости, тож були додані до спостережень.

У таблиці 2.1 знаходяться дані поступового росту рослини: центрального, лівого та правого ростків.

День	1	2	3	4	5	6	7	8	9	10	11	12	13
См(центр)	8	8.5	9	10.5	12.5	17.5	24.5	35	45.5	50	54	57	60
См(лівий)	-	-	-	-	-	-	-	3.5	5	6.5	10	12.5	16.5
См(правий)	-	-	-	-	-	-	-	2	2.5	3	3.5	4	5.5

Таблиця 2.1 Ріст рослини у теплиці

Дивіться поступові фото росту рослини у додатку Б.

4.5.1 Макет теплиці з макетною платою

Був побудований макет теплиці. Зверху знаходиться кришка, яка має всередині відділення для електроніки. Це необхідно для того, щоб відмежувати електроніку та вологе середовище. Спереду, за передньою кришкою, знаходиться оргскло, яке можна витягнути через верхню кришку. Дане скло необхідне для того, щоб перевірити стан рослини, не випустивши при цьому тепле повітря.

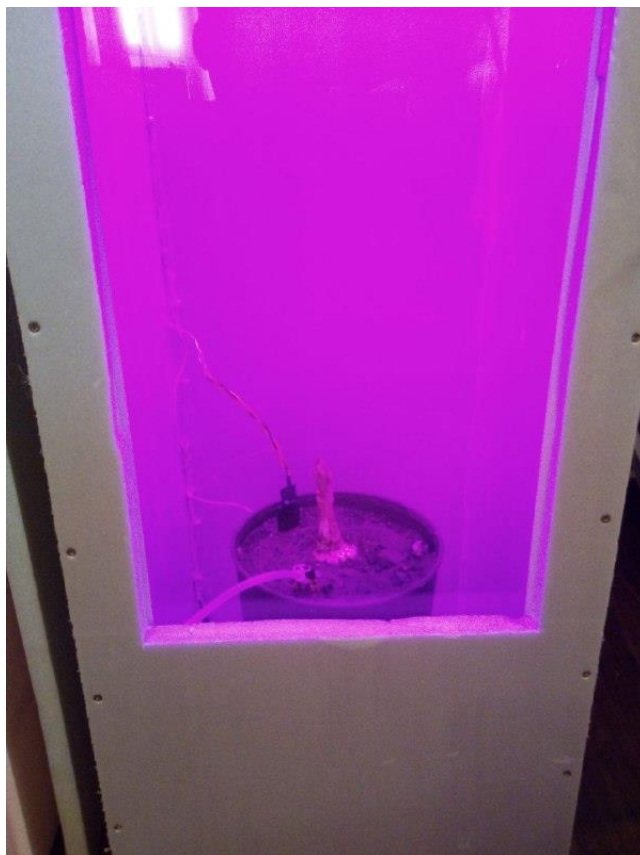


Рисунок 4.29 – Працюючий макет теплиці

Більше ілюстрацій макету дивіться у додатку В.

Була розроблена також принципова схема. За нею була створена макетна плата для перевірки працездатності схеми. Вона та уся електроніка, окрім датчиків та деяких виконавчих елементів, знаходиться у верхній кришці.

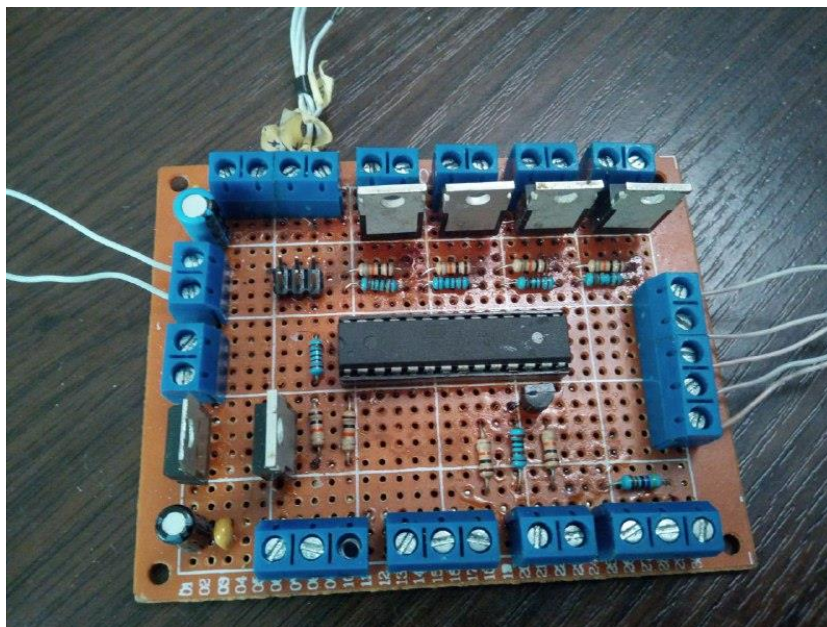


Рисунок 4.30 – Макетна плата

4.5.2 Додаток для ОС Windows

Для налаштування роботи мікротеплиці був створений додаток для ОС Windows. Комп'ютер з даним додатком під'єднується до плати завдяки USB порту. Між платою та комп'ютером знаходиться USB-UART / USB-TTL перетворювач задля уможливлення спілкування мікроконтролера з комп'ютером.

Greenhouse control

Device:

Date Time:
 Date: 16.6.19
 Time: 13:4:15

New Date Time:
 New date:
 New time:

Light:
 Timescale:
 New begin:
 New end:
 Status: Enable

Air:
 Timescale:
 New begin:
 New end:
 Status: Enable

Temperature:
 Temp: 25 °C
 New temp: °C
 Temp (top): 26 °C
 Temp (dwn): 25 °C
 Heating: Disable

Humidity:
 Humidity: 640
 New humidity:
 Humidity (grnd): 512
 Humidity (air): 73 %
 Water at bottom: Yes
 Watering: Disable

Рисунок 4.31 – Додаток для налаштування мікротеплиці

4.6 Варіант подальшого розвитку роботи

Для повноцінної багатофункціональної теплиці з повним спектром можливостей можна додати охолоджуючий елемент для забезпечення працездатності у спекотних країнах, або при спекотному навколишньому середовищі.

Також дуже актуальним буде створення мобільного додатку для ОС Android та IOS. Це спростить налаштування мікротеплиці, адже не доведеться тягнути дроти через усе приміщення або ходити з ноутбуком у руках.

Необхідно відзначити, що також можливо обклеїти зсередини стінки теплиці світло відражаючим матеріалом. Це дозволить усунути поглинання стінками світла та заощадить споживання електроенергії.

Як один з напрямків розвитку можливе також впровадження штучного інтелекту, що буде аналізувати стан рослини, на основі нейронних мереж вирішувати коли поливати рослину і чи потрібен їй негайний догляд з боку людини.

Загалом, напрямок розвитку автоматизованих теплиць є дуже перспективним, адже чисельність населення збільшується і тим самим збільшується потреба у кількості їжі. Окрім того, у вік цифрових технологій усе менше людей хочуть бути аграрниками та фермерами – людство стрімко йде до повної автоматизації праці.

Тож дана тема є дуже перспективною для вивчення та подальшого вдосконалення.

					IA52.200БАК.005 ПЗ	Лист
						57
Ізм.	Лист	№ докум.	Підпис	Дата		

4.7 Висновки

Були спроектовані принципова, функціональна та структурна схеми, а також складальний кресленик та друкована плата. Також була змодельована 3-д модель плати. Після цього була реалізована макетна плата для тестування працездатності схеми.

Була створена програма для мікроконтролера, яка здійснює керування теплицею, а також додаток на ОС Windows, за допомогою якого можна налаштовувати роботу теплиці.

Загалом була розроблена автоматизована система керування мікротеплицею, з урахуванням усіх потреб рослин. Були розраховані необхідні параметри освітлення та підібрана необхідна світлодіодна стрічка.

					ІА52.200БАК.005 ПЗ	Лист
						58
Ізм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

Була створена автоматизована система керування мікротеплицею, яка задовольняє усі потреби рослин.

Для цього були досліджені основні фактори успішного росту рослин. Після цього були створені принципова, функціональна та структурна схеми, а також складальний кресленик та друкована плата. Також була змодельована 3-д модель плати. Після цього була реалізована макетна плата для тестування працездатності схеми.

Готова автоматизована система управління здатна тривалий час підтримувати життєдіяльність рослини без втручання людини.

Даний напрямок досліджень дуже перспективний, адже все в наш час намагаються автоматизувати і люди стають більш схильні до урбанізації, ніж до фермерства.

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		59

ПЕРЕЛІК ПОСИЛАНЬ

1. DS18B20 - Programmable Resolution 1-Wire Digital Thermometer - Maxim Integrated Products, Inc., 2018
2. IRLZ44N Datasheet – Режим доступу: <http://www.irf.com/product-info/datasheets/data/irlz44n.pdf/>.
3. BC547 Datasheet – Режим доступу: <https://www.sparkfun.com/datasheets/Components/BC546.pdf/>.
4. DS1302 Trickle Charge Timekeeping Chip – Dallas Semiconductors Inc.
5. ATmega328P Datasheet – ATMEL Inc,
6. Richard H. Barnett, Sarah Cox, Larry O'Cull. Embedded C Programming and the Atmel AVR 2nd Edition.
7. Mirosław Kardas . AVR Microcontrollers C – Programming Basics.
8. Programming With AVR Microcontroller – Research design lab
9. Сост. Ю.А. Шпак. Программирование на языке C для AVR и PIC микроконтроллеров – К.: «МК-Пресс», М: Издательский дом «Додэка-XXI» 2007. – 400 с., ил.
10. Белов А. В. Разработка устройств на микроконтроллерах AVR - шагаем от «чайника» до профи.
11. Михаил Фленов. Библия C# 2-е издание – 560 с, 2011
12. Бен Ватсон . C# 4.0 на примерах – 608 с, 2011
13. DHT22 Datasheet - Digital-output relative humidity & temperature sensor/module - Aosong Electronics Co.,Ltd

					IA52.200БАК.005 ПЗ	Лист
						60
Ізм.	Лист	№ докум.	Підпис	Дата		

14. Needings of light by plants – Режим доступу:

<https://лампочка.com/articles/skolko-sveta-neobkhodimo-rasteniyam/>

15. Light for plant in fito-spectre – Режим доступу:

<https://mirfermera.ru/444-iskusstvennoe-osveschenie-dlya-komnatnyh-rasteniy.html/>.

					IA52.200БАК.005 ПЗ	Лист
						61
Ізм.	Лист	№ докум.	Підпис	Дата		

ДОДАТОК А

Лістинг розроблених програм

Програма для мікроконтролера

```
#define F_CPU 8000000UL //16MHz

#include <stdio.h>
#include <stdlib.h>
#include <avr/io.h>
#include <util/delay.h>

#include <avr/interrupt.h>

#include <avr/eeprom.h>

// #define DATA_REGISTER_EMPTY (1<<UDRE)
// #define RX_COMPLETE (1<<RXC)
// #define FRAMING_ERROR (1<<FE)
// #define PARITY_ERROR (1<<UPE)
// #define DATA_OVERRUN (1<<DOR)

// USART Receiver buffer
#define RX_BUFFER_SIZE 64
unsigned char rx_buffer[RX_BUFFER_SIZE];

unsigned char rx_wr_index,rx_rd_index,rx_counter;

char read_enable = 0;
```

					ІА52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		62


```

void init_uart()
{
    // setting the baud rate based on the data sheet
    UBRR0H = 0x00; //(unsigned char) ( uValue>> 8); // 0x00
    //UBRR0L = 0x67; //(unsigned char) uValue; // 0x0C // for 16 MHz
    UBRR0L = 0x33; // for 8 MHz
    // enabling TX & RX
    //UCSR0B = (1 << RXEN0) | (1 << TXEN0);
    //UCSR0A = (1 << UDRE0) | (1 << U2X0);
    //UCSR0C = 0x06; // Set frame: 8data, 1 stop

    UCSR0A = 0x00;
    // enabling TX & RX
    UCSR0B = (1 << RXCIE0) | (1 << RXEN0) | (1 << TXEN0); //(1 <<
RXEN0) | (1 << TXEN0);
    //UCSR0A = (1 << UDRE0) | (1 << U2X0);
    UCSR0C = 0x86; // Set frame: 8data, 1 stop
}

void USART_Transmit(uint8_t data)
{
    /* Wait for empty transmit buffer */
    while (!(UCSR0A & (1 << UDRE0)));

    /* Put data into buffer, sends the data */
    UDR0 = data;
}

void USART_PrintUInt(int len, char *mes, unsigned int data)

```

					IA52.200БАК.005 ПЗ	Лист
						63
Ізм.	Лист	№ докум.	Підпис	Дата		

```

{
    for(int i = 0; i < len; i++)
    {
        USART_Transmit(mes[i]);
    }

    char s[20];

    //utoa(data, s, 10);
    sprintf(s, "%u", data);

    for(int i = 0; i < 20; i++)
    {
        if(s[i] == 0)
        {
            break;
        }
        USART_Transmit(s[i]);
    }
}
/*
void USART_PrintInt(int len, char *mes, unsigned int data)
{
    for(int i = 0; i < len; i++)
    {
        USART_Transmit(mes[i]);
    }

    char s[20];

```

```

//utoa(data, s, 10);
sprintf(s, "%d", data);

for(int i = 0; i < 20; i++)
{
    if(s[i] == 0)
    {
        break;
    }
    USART_Transmit(s[i]);
}
*/

void IntegertoString(char * string, int number) {

    if(number == 0) { string[0] = '0'; return; };
    int divide = 0;
    int modResult;
    int length = 0;
    int isNegative = 0;
    int copyOfNumber;
    //int offset = 0;
    copyOfNumber = number;
    if( number < 0 ) {
        isNegative = 1;
        number = 0 - number;
        length++;
    }
    while(copyOfNumber != 0)

```

					IA52.200БАК.005 ПЗ	Лист
						65
Ізм.	Лист	№ докум.	Підпис	Дата		

```

{
    length++;
    copyOfNumber /= 10;
}

for(divide = 0; divide < length; divide++) {
    modResult = number % 10;
    number = number / 10;
    string[length - (divide + 1)] = modResult + '0';
}
if(isNegative) {
    string[0] = '-';
}
string[length] = '\0';
}

void USART_PrintInt(int data)
{
    char s[20];

    char *mess = s;

    IntegertoString(s, data);

    while(*mess != 0)
    {
        USART_Transmit(*mess);
        mess++;
    }
}

```

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		66

```

    }
}

void USART_PrintUChar(int len, char *mes, unsigned char data)
{
    for(int i = 0; i < len; i++)
    {
        USART_Transmit(mes[i]);
    }

    char s[20];

    //utoa(data, s, 10);
    sprintf(s, "%i", data);

    for(int i = 0; i < 20; i++)
    {
        if(s[i] == 0)
        {
            break;
        }
        USART_Transmit(s[i]);
    }
}

void USART_PrintText(int len, char *mes)
{
    for(int i = 0; i < len; i++)
    {

```

```

        USART_Transmit(mes[i]);
    }
}

void USART_PrintTextLn(char *mes)
{
    while(*mes != 0)
    {
        USART_Transmit(*mes);
        mes++;
    }
}

ISR(USART_RX_vect )
{
    char status;
    unsigned char data;
    status=UCSR0A;
    data=UDR0;

    if (status & (1<<RXC0))
    {
        if (data == 210)
        {
            rx_wr_index=0;
            read_enable = 1;
        }
        else if ((data == 230) && (read_enable == 1))
        {
            cli();

```

					IA52.200БАК.005 ПЗ	Лист
						68
Ізм.	Лист	№ докум.	Підпис	Дата		

```

read_enable = 0;
//lcd_out = 1;

rx_buffer[rx_wr_index++] = '\0';

switch (rx_buffer[0])
{
    case 201:
        for(int i = 0; i < rx_wr_index; i++)
        {
            eeprom_write_byte((uint8_t*)100 + i,
rx_buffer[i + 1]);
        }
        break;

    case 202:
        for(int i = 0; i < rx_wr_index; i++)
        {
            eeprom_write_byte((uint8_t*)200 + i,
rx_buffer[i + 1]);
        }
        break;

    case 203:
        eeprom_write_byte((uint8_t*)300,
rx_buffer[1]);

        break;

    case 204:

```

					IA52.200БАК.005 ПЗ	Лист
						69
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        for(int i = 0; i < rx_wr_index; i++)
        {
            eeprom_write_byte((uint8_t*)400 + i,
rx_buffer[i + 1]);
        }
        break;

    case 205:
        for(int i = 0; i < rx_wr_index; i++)
        {
            eeprom_write_byte((uint8_t*)500 + i,
rx_buffer[i + 1]);
        }
        break;

    default:
    {
        break;
    }
}

sei();
}
else if (read_enable == 1)
{
    rx_buffer[rx_wr_index++]=data;
    if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;

    if (++rx_counter == RX_BUFFER_SIZE)
    {

```



```

        rx_counter=0;
        //rx_buffer_overflow=1;
    }

}

}
}

#ifndef UART_H_
#define UART_H_

void init_uart();

void USART_Transmit(uint8_t data);

void USART_PrintUInt(int len, char *mes, unsigned int data);

void USART_PrintInt(int data);

void USART_PrintUChar(int len, char *mes, unsigned char data);

void USART_PrintText(int len, char *mes);

void USART_PrintTextLn(char *mes);

#endif /* UART_H_ */

#ifndef DS1820_H_
#define DS1820_H_

int DS18B20_ReadTemperature();

```

					IA52.200БАК.005 ПЗ	Лист
						71
Ізм.	Лист	№ докум.	Підпис	Дата		

```
#endif /* DS1820_H_ */

#include <stdlib.h>
#include <avr/io.h>
#include <util/delay.h>

#define OW_PORT      PORTB
#define OW_DDR      DDRB
#define OW_PIN      PINB
#define OW_PINX      0

void owi_WriteBit1()
{

    OW_DDR |= 1 << OW_PINX;
    OW_PORT &= ~(1 << OW_PINX);
    _delay_us(6);

    OW_DDR &= ~(1<<OW_PINX);
    OW_PORT |= (1<<OW_PINX);
    _delay_us(64);
}

void owi_WriteBit0()
{

    OW_DDR |= 1 << OW_PINX;
    OW_PORT &= ~(1 << OW_PINX);
    _delay_us(60);
```

```

OW_DDR &= ~(1<<OW_PINX);
OW_PORT |= (1<<OW_PINX);
_delay_us(10);

}

unsigned char owi_ReadBit()
{
    unsigned char bitsRead;

    OW_DDR |= 1 << OW_PINX;
    OW_PORT &= ~(1 << OW_PINX);
    _delay_us(6);

    OW_DDR &= ~(1<<OW_PINX);
    OW_PORT |= (1<<OW_PINX);
    _delay_us(9);

    bitsRead = OW_PIN & (1 << OW_PINX);
    _delay_us(55);

    return bitsRead;
}

void OwSendByte(unsigned char data)
{
    unsigned char tmp;
    unsigned char i;

```

```

// do once for each bit
for(i = 0; i < 8; i++)
{
    tmp = data & 0x01;
    if(tmp) owi_WriteBit1();
    else owi_WriteBit0();

    data >>= 1;
}
}

unsigned char OwReceiveByte()
{
    unsigned char data;
    unsigned char i;

    // clear the temporary input variable
    data = 0x00;

    // do once for each bit
    for(i = 0; i < 8; i++)
    {
        // shift temporary input variable right
        data >>= 1;

        // set the msb if a '1' value is read from the bus
        // leave as it is '0' else
        if(owi_ReadBit()) data |= 0x80;
    }
}

```

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		74

```

return data;

}

unsigned char OwReset()
{
    unsigned char detected;

    OW_DDR |= 1 << OW_PINX;
    OW_PORT &= ~(1 << OW_PINX);

    _delay_us(480);

    OW_DDR &= ~(1 << OW_PINX);
    OW_PORT |= (1 << OW_PINX);
    _delay_us(70);

    detected = ((~OW_PIN) & (1 << OW_PINX));
    _delay_us(410);

    return detected;
}

int DS18B20_ReadTemperature()
{
    unsigned char temp[2];

    int digit;

```

					IA52.200БАК.005 ПЗ	Лист
						75
Ізм.	Лист	№ докум.	Підпис	Дата		

```

//int decimal;

// reset
OwReset();

// send skip addr
OwSendByte(0xCC);

// send start conversion command
OwSendByte(0x44);

// wait until conversion is finished
// bus line is held low until conversion is finished
while(!owi_ReadBit());

// reset
OwReset();

// send skip addr
OwSendByte(0xCC);

// send read scratchpad command
OwSendByte(0xBE);

// read only first two bytes (temp low, temp high)
// and place then in the 16 bit temperature variable
temp[0] = OwReceiveByte();
temp[1] = OwReceiveByte();

//Store temperature integer digits and decimal digits

```

					IA52.200БАК.005 ПЗ	Лист
						76
Ізм.	Лист	№ докум.	Підпис	Дата		

```
digit = temp[0] >> 4;
digit |= (temp[1] & 0x7) << 4;
```

```
//Store decimal digits
```

```
//decimal = temp[0] & 0xF;
```

```
//decimal *= 625;
```

```
//sprintf(buffer, "\r\n%+d.%01u C", digit, decimal);
```

```
return digit;
```

```
}
```

```
#ifndef DS1302_H_
```

```
#define DS1302_H_
```

```
void write(unsigned char cmd);
```

```
void end_write_data();
```

```
unsigned char read();
```

```
void SetData(unsigned int hour, unsigned int min, unsigned int sec,
unsigned int date, unsigned int mon, unsigned int year);
```

```
unsigned char GetSec();
```

```
unsigned char GetMin();
```

```
unsigned char GetHour();
```

					IA52.200БАК.005 ПЗ	Лист
						77
Ізм.	Лист	№ докум.	Підпис	Дата		

```
unsigned char GetDate();

unsigned char GetMon();

unsigned char GetYear();

#endif /* DS1302_H_ */

#define F_CPU 8000000UL //16MHz

#include <avr/io.h>
#include <util/delay.h>

//#define F_CPU 4000000

#define SCLK 7
#define IO 6
#define E 5
#define DDR_RTC DDRD
#define PORT_RTC PORTD
#define PIN_RTC PIND

//посылаем команду или байт данных в часы
void write(unsigned char cmd)
{
    DDR_RTC |= (1<<E) | (1<<SCLK);
    PORT_RTC |= (1<<E);//E=1
    _delay_us(4);
    DDR_RTC |= (1<<IO);//ВЫХОД
    for(unsigned char i=0; i<8; i++)
```



```

{
    if((cmd&(1<<i)) == 1<<i)
    {
        PORT_RTC |= (1<<IO);
    }
    else
    {
        PORT_RTC &= ~(1<<IO);
    }
    PORT_RTC |= (1<<SCLK);
    _delay_us(1);
    PORT_RTC &= ~(1<<IO);
    PORT_RTC &= ~(1<<SCLK);
}
}

//вызываем после записи байта данных в часы
void end_write_data()
{
    PORT_RTC &= ~(1<<E);
}

unsigned char read()
{
    unsigned char readbyte=0;
    DDR_RTC &= ~(1<<IO);
    for(unsigned char i=0;i<8;i++)
    {
        PORT_RTC |= 1<<SCLK;
        if((PIN_RTC & (1<<IO))==0)
        {

```

```

        readbyte &= ~(1<<i);
    }
    else
    {
        readbyte |= 1<<i;
    }
    _delay_us(10);
    PORT_RTC &= ~(1<<SCLK);
    _delay_us(2);
}
PORT_RTC &= ~(1<<E);
_delay_us(4);
return readbyte;
}

```

```

void SetData(unsigned char hour, unsigned char min, unsigned char sec,
unsigned char date, unsigned char mon, unsigned char year)

```

```

{
    //set hour
    write(0x84);
    write(((hour / 10) << 4) | (hour % 10));
    end_write_data();

    //set minutes
    write(0x82);
    write(((min / 10) << 4) | (min % 10));
    end_write_data();

    //set seconds
    write(0x80);

```

					IA52.200БАК.005 ПЗ	Лист
						80
Ізм.	Лист	№ докум.	Підпис	Дата		

```

write(((sec / 10) << 4) | (sec % 10));
end_write_data();

//set date
write(0x86);
write(((date / 10) << 4) | (date % 10));

end_write_data();

//set mon
write(0x88);
write(((mon / 10) << 4) | (mon % 10));
end_write_data();

//set year
write(0x8C);
write(((year / 10) << 4) | (year % 10));
end_write_data();
}

unsigned char GetHour()
{
    unsigned char hour, hour10;
    write(0x85);
    hour = read();
    hour10 = ((hour >> 4) & 0x11);
    hour &= 0x0F;
    return (hour10 * 10) + hour;
}

```

					IA52.200БАК.005 ПЗ	Лист
						81
Ізм.	Лист	№ докум.	Підпис	Дата		

```
unsigned char GetMin()
{
    unsigned char min, min10;
    write(0x83);
    min = read();
    min10 = ((min>>4) & 0b000000111);
    min &= 0x0F;
    return (min10 * 10) + min;
}
```

```
unsigned char GetSec()
{
    unsigned char sec, sec10;
    write(0x81);
    sec = read();
    sec10 = ((sec>>4) & 0b000000111);
    sec &= 0x0F;
    return (sec10 * 10) + sec;
}
```

```
unsigned char GetDate()
{
    unsigned char date, date10;
    write(0x87);
    date = read();
    date10 = ((date>>4) & 0x03);
    date &= 0x0F;
    return (date10 * 10) + date;
}
```

```
unsigned char GetMon()
{
    unsigned char month, month10;
    write(0x89);

    month = read();
    month10 = ((month>>4) & 0x01);
    month &= 0x0F;
    return (month10 * 10) + month;
}
```

```
unsigned char GetYear()
{
    unsigned char year, year10;
    write(0x8D);
    year = read();
    year10 = ((year>>4) & 0x0F);
    year &= 0x0F;
    return (year10 * 10) + year;
}
```

```
#include <avr/io.h>
#define F_CPU 8000000UL //16MHz
#include <util/delay.h>
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <avr/eeprom.h>
```

```
#include <string.h>
```

```
#include <avr/interrupt.h>
```

```
#include <util/atomic.h>
```

```
#include "UART.h"
```

```
#include "DS1302.h"
```

```
#include "DS1820.h"
```

```
#include "DHT.h"
```

```
#include "AnalogSensors.h"
```

```
ISR (TIMER1_COMPA_vect)
```

```
{  
    cli();
```

```
    uint8_t datebuffer[100];
```

```
    uint8_t turn_on_off = 0;
```

```
    uint8_t times_count = 0;
```

```
  
    // Read DateTime
```

```
    uint8_t sec = GetSec();
```

```
    uint8_t min = GetMin();
```

```
    uint8_t hour = GetHour();
```

```
    uint8_t day = GetDate();
```

```
    uint8_t mon = GetMon();
```

```
uint8_t year = GetYear();
```

```
datebuffer[0] = 210;
```

```
datebuffer[1] = 205;
```

```
datebuffer[2] = sec;
```

```
datebuffer[3] = min;
```

```
datebuffer[4] = hour;
```

```
datebuffer[5] = day;
```

```
datebuffer[6] = mon;
```

```
datebuffer[7] = year;
```

```
datebuffer[8] = 230;
```

```
USART_PrintText(9, datebuffer);
```

```
// Turn on/off light
```

```
turn_on_off = 0;
```

```
times_count = eeprom_read_byte((uint8_t*)100);
```

```
//sprintf(datebuffer, "\nLight Count: %d", times_count);
```

```
//USART_PrintText(strlen(datebuffer), datebuffer);
```

```
datebuffer[0] = 210;
```

```
datebuffer[1] = 201;
```

```
datebuffer[2] = times_count;
```

```
for(uint8_t i = 0; i < times_count; i++)
```

```
{
```

```
    uint8_t beg_h = eeprom_read_byte((uint8_t*)(101 + (i * 4)));
```

```
    uint8_t beg_m = eeprom_read_byte((uint8_t*)(102 + (i * 4)));
```

```
    uint8_t end_h = eeprom_read_byte((uint8_t*)(103 + (i * 4)));
```

					IA52.200БАК.005 ПЗ	Лист
						85
Ізм.	Лист	№ докум.	Підпис	Дата		

```

uint8_t end_m = eeprom_read_byte((uint8_t*)(104 + (i * 4)));

if(min >= beg_m && min <= end_m && hour >= beg_h && hour
<= end_h)
{
    turn_on_off = 1;
}

datebuffer[3 + (i * 4)] = beg_h;
datebuffer[4 + (i * 4)] = beg_m;
datebuffer[5 + (i * 4)] = end_h;
datebuffer[6 + (i * 4)] = end_m;

//sprintf(datebuffer, "\nLight BEG-END: %d:%d-%d:%d", beg_h,
beg_m, end_h, end_m);
//USART_PrintText(strlen(datebuffer), datebuffer);
}

if(turn_on_off)
{
    // Turn on light
    PORTB |= 1 << 3;

    //sprintf(datebuffer, "\nLight was turned on");
    //USART_PrintText(strlen(datebuffer), datebuffer);
}
else
{
    // Turn off light
    PORTB &= ~(1 << 3);
}

```

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		86


```

        //sprintf(datebuffer, "\nLight was turned off");
        //USART_PrintText(strlen(datebuffer), datebuffer);
    }

```

```

datebuffer[3 + (times_count * 4)] = turn_on_off;
datebuffer[4 + (times_count * 4)] = 230;
USART_PrintText(5 + (times_count * 4), datebuffer);

```

```

// Turn on/off air
turn_on_off = 0;
times_count = eeprom_read_byte((uint8_t*)200);

```

```

datebuffer[0] = 210;
datebuffer[1] = 202;
datebuffer[2] = times_count;

```

```

//sprintf(datebuffer, "\nAir Count: %d", times_count);
//USART_PrintText(strlen(datebuffer), datebuffer);

```

```

for(uint8_t i = 0; i < times_count; i++)
{
    int beg_h = eeprom_read_byte((uint8_t*)(201 + (i * 4)));
    int beg_m = eeprom_read_byte((uint8_t*)(202 + (i * 4)));
    int end_h = eeprom_read_byte((uint8_t*)(203 + (i * 4)));
    int end_m = eeprom_read_byte((uint8_t*)(204 + (i * 4)));
}

```

```

        if(min >= beg_m && min <= end_m && hour >= beg_h && hour
        <= end_h)
        {
            turn_on_off = 1;
        }

        datebuffer[3 + (i * 4)] = beg_h;
        datebuffer[4 + (i * 4)] = beg_m;
        datebuffer[5 + (i * 4)] = end_h;
        datebuffer[6 + (i * 4)] = end_m;

        //sprintf(datebuffer, "\nAir BEG-END: %d:%d-%d:%d", beg_h,
        beg_m, end_h, end_m);
        //USART_PrintText(strlen(datebuffer), datebuffer);
    }

    if(turn_on_off)
    {
        // Turn on air
        PORTB |= 1 << 5;

        //sprintf(datebuffer, "\nAir was turned on");
        //USART_PrintText(strlen(datebuffer), datebuffer);
    }
    else
    {
        // Turn off air
        PORTB &= ~(1 << 5);

        //sprintf(datebuffer, "\nAir was turned off");
    }

```

					IA52.200БАК.005 ПЗ	Лист
						88
Ізм.	Лист	№ докум.	Підпис	Дата		

```
        //USART_PrintText(strlen(datebuffer), datebuffer);  
    }  
  
    datebuffer[3 + (times_count * 4)] = turn_on_off;  
    datebuffer[4 + (times_count * 4)] = 230;  
    USART_PrintText(5 + (times_count * 4), datebuffer);  
  
    // Turn on power  
    PORTD |= 1 << 3;  
  
    //USART_PrintTextLn("\nPower turned on");  
  
    // Read temperature DS1820  
    uint8_t temp_DS1820 = DS18B20_ReadTemperature();  
  
    datebuffer[0] = 210;  
    datebuffer[1] = 203;  
    datebuffer[2] = temp_DS1820;  
  
    //sprintf(datebuffer,          "\nTemperatureDS1820          -          %d",  
    DS18B20_ReadTemperature());  
    //USART_PrintTextLn(datebuffer);  
  
    // Read temperature and humidity DHT22
```

					IA52.200БАК.005 ПЗ	Лист
						89
Ізм.	Лист	№ докум.	Підпис	Дата		

```

uint8_t temp_DHT22 = 0;
unsigned int humid_DHT22 = 0;
dht_gettemperaturehumidity(&temp_DHT22, &humid_DHT22);

//sprintf(datebuffer, "\nTemperatureDHT - %d\nHumidityDHT - %d",
temp_DHT22, humid_DHT22);
//USART_PrintTextLn(datebuffer);

datebuffer[3] = temp_DHT22;

// Turn on/off heating
int temp = eeprom_read_byte((uint8_t*)300);

//sprintf(datebuffer, "\nTemper - %d", temp);
//USART_PrintTextLn(datebuffer);

if(temp_DS1820 < temp && temp_DHT22 < temp)
{
    // Turn on
    PORTB |= 1 << 2;

    datebuffer[4] = 1;

    //sprintf(datebuffer, "\nHeating was turned on");
    //USART_PrintText(strlen(datebuffer), datebuffer);
}
else if(temp_DS1820 > temp || temp_DHT22 > temp)
{
    // Turn off

```

					IA52.200БАК.005 ПЗ	Лист
						90
Ізм.	Лист	№ докум.	Підпис	Дата		

```
PORTB &= ~(1 << 2);
```

```
datebuffer[4] = 0;
```

```
//sprintf(datebuffer, "\nHeating was turned off");
```

```
//USART_PrintText(strlen(datebuffer), datebuffer);
```

```
}
```

```
datebuffer[5] = 230;
```

```
USART_PrintText(6, datebuffer);
```

```
// Check humidity
```

```
ADC_Init(1);
```

```
int humidSens = getAnalogValue();
```

```
datebuffer[0] = 210;
```

```
datebuffer[1] = 204;
```

```
//datebuffer[2] = eeprom_read_byte((uint8_t*)401);
```

```
int hum = humidSens;
```

```
uint8_t c = 1;
```

```
while(hum > 9)
```

```
{
```

```
    hum /= 10;
```

```
    c++;
```

```
}
```

```
datebuffer[2] = c;
```

```
int num = humidSens;
```

					IA52.200БАК.005 ПЗ	Лист
						91
Ізм.	Лист	№ докум.	Підпис	Дата		

```

for(uint8_t i = c; i > 0; i--)
{
    datebuffer[2 + i] = num % 10;

    num /= 10;
}

//sprintf(datebuffer, "\nHumidity - %d", humidSens);
//USART_PrintText(strlen(datebuffer), datebuffer);


// Check water and turn on/off watering
// Check water
if(PIND & (1<<4))
{
    //sprintf(datebuffer, "\nVody net");

    int humid = eeprom_read_byte((uint8_t*)400);

    //sprintf(datebuffer, "\nHumid - %d", humid);
    //USART_PrintTextLn(datebuffer);
    datebuffer[8] = 0;

    if(humidSens >= humid)
    {
        // Turn on watering
        PORTB |= 1 << 4;

        //sprintf(datebuffer, "\nKlap was turned on");
    }
}

```

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		92

```

        //USART_PrintText(strlen(datebuffer), datebuffer);

        datebuffer[9] = 1;
    }
}
else
{
    //sprintf(datebuffer, "\nVoda est");
    datebuffer[8] = 1;

    // Turn off watering
    PORTB &= ~(1 << 4);
    datebuffer[9] = 0;

    //sprintf(datebuffer, "\nKlap was turned off");
    //USART_PrintText(strlen(datebuffer), datebuffer);
}
USART_PrintText(10, datebuffer);


// Turn off power
//PORTD &= ~(1 << 3);

//USART_PrintTextLn("\nPower turned off");

//USART_PrintTextLn("\n");

sei();

```

					IA52.200БАК.005 ПЗ	Лист
						93
Ізм.	Лист	№ докум.	Підпис	Дата		

```

    TCNT1 = 0;
}

void timer_init()
{
    TCCR1B |= (1 << WGM12) | (1 << CS12) | (1 << CS10); // CTC mode,
Clock/1024
    TCNT1=0x00; //здесь увеличиваются тики

    //OCR1A=7813; //записываем число в регистр сравнения

    OCR1A = 65535;

    TIMSK1 |= (1 << OCIE1A); // Enable the compare match interrupt
    sei(); // Now enable global interrupts
}

int main(void)
{
    init_uart();

    if(eeprom_read_byte((uint8_t*)0) != (uint8_t)17)
    {
        eeprom_write_byte((uint8_t*)0, 17);

        eeprom_write_byte((uint8_t*)100, 1);
        eeprom_write_byte((uint8_t*)101, 6);
        eeprom_write_byte((uint8_t*)102, 0);
        eeprom_write_byte((uint8_t*)103, 22);
    }
}

```

					IA52.200БАК.005 ПЗ	Лист
						94
Ізм.	Лист	№ докум.	Підпис	Дата		


```

        eeprom_write_byte((uint8_t*)104, 0);

        eeprom_write_byte((uint8_t*)200, 0);

        eeprom_write_byte((uint8_t*)300, 30);

        eeprom_write_byte((uint8_t*)400, 200);

        //USART_PrintTextLn("\nSettings initialized by default");
    }

    // Nalichie Vody
    DDRD &= ~(1 << 4);

    // Nagrev
    DDRB |= 1 << 2;

    // Light
    DDRB |= 1 << 3;

    // Klap
    DDRB |= 1 << 4;

    // Vozd
    DDRB |= 1 << 5;

    // Pitanie datchikov
    DDRD |= 1 << 3;

    _delay_ms(3000);

```

					IA52.200БАК.005 ПЗ	Лист
						95
Ізм.	Лист	№ докум.	Підпис	Дата		

```

timer_init();

while(1)
{
    //uint8_t t = cycle_num;
    //eeprom_write_byte((uint8_t*)46, t);
}
}

#ifndef DHT_H_
#define DHT_H_

#include <stdio.h>
#include <avr/io.h>

//setup port
#define DHT_DDR DDRD
#define DHT_PORT PORTD
#define DHT_PIN PIND
#define DHT_INPUTPIN 2

//timeout retries
#define DHT_TIMEOUT 200

int8_t dht_gettemperature(int *temperature);
int8_t dht_gethumidity(int *humidity);
int8_t dht_gettemperaturehumidity(int *temperature, int *humidity);

#endif

```

					IA52.200БАК.005 ПЗ	Лист
Ізм.	Лист	№ докум.	Підпис	Дата		96

```
#define F_CPU 8000000UL //16MHz

#include <stdio.h>
#include <string.h>
#include <avr/io.h>
#include <util/delay.h>

#include "dht.h"

/*
 * get data from sensor
 */

int8_t dht_getdata(int *temperature, int *humidity)
{
    uint8_t bits[5];
    uint8_t i,j = 0;

    memset(bits, 0, sizeof(bits));

    //reset port
    DHT_DDR |= (1<<DHT_INPUTPIN); //output
    DHT_PORT |= (1<<DHT_INPUTPIN); //high
    _delay_ms(100);

    //send request
    DHT_PORT &= ~(1<<DHT_INPUTPIN); //low
    _delay_us(500);
    DHT_PORT |= (1<<DHT_INPUTPIN); //high
    DHT_DDR &= ~(1<<DHT_INPUTPIN); //input
```

```

    _delay_us(40);

    //check start condition 1
    if((DHT_PIN & (1<<DHT_INPUTPIN))) {
        return -1;
    }
    _delay_us(80);
    //check start condition 2
    if(!(DHT_PIN & (1<<DHT_INPUTPIN))) {
        return -1;
    }
    _delay_us(80);

    //read the data
    uint16_t timeoutcounter = 0;
    for (j=0; j<5; j++) { //read 5 byte
        uint8_t result=0;
        for(i=0; i<8; i++) { //read every bit
            timeoutcounter = 0;
            while(!(DHT_PIN & (1<<DHT_INPUTPIN))) { //wait for an
high input (non blocking)
                timeoutcounter++;
                if(timeoutcounter > DHT_TIMEOUT) {
                    return -1; //timeout
                }
            }
            _delay_us(30);
            if(DHT_PIN & (1<<DHT_INPUTPIN)) //if input is high
after 30 us, get result
                result |= (1<<(7-i));

```

					IA52.200БАК.005 ПЗ	Лист
						98
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        timeoutcounter = 0;
        while(DHT_PIN & (1<<DHT_INPUTPIN)) { //wait until
input get low (non blocking)
            timeoutcounter++;
            if(timeoutcounter > DHT_TIMEOUT) {
                return -1; //timeout
            }
        }
    }
    bits[j] = result;
}

//reset port
DHT_DDR |= (1<<DHT_INPUTPIN); //output
DHT_PORT |= (1<<DHT_INPUTPIN); //low
_delay_ms(100);

//check checksum
if ((uint8_t)(bits[0] + bits[1] + bits[2] + bits[3]) == bits[4]) {
    //return temperature and humidity
    uint16_t rawhumidity = bits[0]<<8 | bits[1];
    uint16_t rawtemperature = bits[2]<<8 | bits[3];
    if(rawtemperature & 0x8000) {
        *temperature = ((rawtemperature & 0x7FFF) / 10.0) * -1.0;
    } else {
        *temperature = (rawtemperature)/10.0;
    }
    *humidity = (rawhumidity)/10.0;
    return 0;
}

```

					IA52.200БАК.005 ПЗ	Лист
						99
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        return -1;
    }

    /*
     * get temperature
     */
    int8_t dht_gettemperature(int *temperature)
    {
        int humidity = 0;

        return dht_getdata(temperature, &humidity);
    }

    /*
     * get humidity
     */
    int8_t dht_gethumidity(int *humidity)
    {
        int temperature = 0;

        return dht_getdata(&temperature, humidity);
    }

    /*
     * get temperature and humidity
     */
    int8_t dht_gettemperaturehumidity(int *temperature, int *humidity)
    {
        //return dht_getdata(temperature, humidity);
    }

```

					IA52.200БАК.005 ПЗ	Лист
						100
Ізм.	Лист	№ докум.	Підпис	Дата		

```
return 0;  
}
```

```
#ifndef ANALOGSENSORS_H_  
#define ANALOGSENSORS_H_
```

```
void ADC_Init(unsigned char pin);  
unsigned int getAnalogValue();
```

```
#endif
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <avr/io.h>
```

```
/* Функция инициализация АЦП */
```

```
void ADC_Init(unsigned char pin)  
{
```

```
    // Включаем АЦП
```

```
    // устанавливаем пределитель преобразователя на 8
```

```
    ADCSRA |= (1 << ADEN) | (1 << ADPS1) | (1 << ADPS0);
```

```
    // устанавливаем пределитель преобразователя на 32
```

```
    //ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS0);
```

```
    // на 128
```

```
    //ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 <<  
ADPS0);
```

					IA52.200БАК.005 ПЗ	Лист
						101
Ізм.	Лист	№ докум.	Підпис	Дата		

```

//выставляем опорное напряжение, как внешний ИОН
// снимать сигнал будем с входа PC0
//ADMUX |= (0 << REFS1) | (1 << REFS0) | (0 << MUX0) | (0 <<
MUX1) | (0 << MUX2) | (0 << MUX3);
ADMUX = 0x00 | pin;
// s1 - 1
}

unsigned int getAnalogValue()
{
    unsigned int res = 0;

    ADCSRA |= (1 << ADSC); // Начинаем преобразование

    //while ((ADCSRA & (1 << ADIF)) == 0); // пока не будет выставлено
флага об окончании преобразования
    while ((ADCSRA & (1 << ADSC)) == 0); // пока не будет выставлено
флага об окончании преобразования

    res = (ADCL | ADCH << 8); // Считываем полученное значение

    return res;
}

```


Програма додатку для ОС Windows

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;

namespace GreenhouseControll
{
    public partial class MainForm : Form
    {
        SerialPort port;
        Thread receiving;

        public MainForm()
        {
            InitializeComponent();
        }

        private void MainForm_Load(object sender, EventArgs e)
        {
            tmrChecker.Enabled = true;
        }
    }
}
```

					IA52.200БАК.005 ПЗ	Лист
						103
Ізм.	Лист	№ докум.	Підпис	Дата		

```
private void MainForm_FormClosing(object sender,
FormClosingEventArgs e)
{
    try
    {
        tmrChecker.Enabled = false;
        if (receiving.IsAlive)
        {
            receiving.Abort();
        }
        if (port.IsOpen)
        {
            port.Close();
        }
    }
    catch
    {
    }
}

private void ReceiveData()
{
    byte[] rx_buffer = new byte[64];
    byte rx_wr_index = 0;
    byte rx_rd_index = 0;
    byte rx_counter = 0;
    byte read_enable = 0;
```

```

while (true)
{
    int data = port.ReadByte();

    if (data == 210)
    {
        rx_wr_index = 0;
        read_enable = 1;
    }
    else if ((data == 230) && (read_enable == 1))
    {
        read_enable = 0;

        switch (rx_buffer[0])
        {
            case 201:
                cbLightTimescale.BeginInvoke(new Action(() =>
                {
                    cbLightTimescale.Items.Clear();
                    for (int i = 0; i < rx_buffer[1]; i++)
                    {
                        cbLightTimescale.Items.Add(rx_buffer[2 + (i *
4)].ToString() + ":" + rx_buffer[3 + (i * 4)].ToString() + " - " +
                        rx_buffer[4 + (i * 4)].ToString() + ":" +
rx_buffer[5 + (i * 4)].ToString());
                    }
                }));
                if (rx_buffer[2 + (rx_buffer[1] * 4)] == 1)
                {

```

					IA52.200БАК.005 ПЗ	Лист
						105
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        lbLightStatus.BeginInvoke(new Action() => {
lbLightStatus.Text = "Enable"; }));
    }
    else
    {
        lbLightStatus.BeginInvoke(new Action() => {
lbLightStatus.Text = "Disable"; }));
    }
    break;

case 202:
    cbAirTimescale.BeginInvoke(new Action() =>
    {
        cbAirTimescale.Items.Clear();
        for (int i = 0; i < rx_buffer[1]; i++)
        {
            cbAirTimescale.Items.Add(rx_buffer[2 + (i *
4)].ToString() + ":" + rx_buffer[3 + (i * 4)].ToString() + " - " +
            rx_buffer[4 + (i * 4)].ToString() + ":" +
rx_buffer[5 + (i * 4)].ToString());
        }
    }));
    if (rx_buffer[2 + (rx_buffer[1] * 4)] == 1)
    {
        lbAirStatus.BeginInvoke(new Action() => {
lbAirStatus.Text = "Enable"; }));
    }
    else
    {

```

```

        lbAirStatus.BeginInvoke(new Action() => {
lbAirStatus.Text = "Disable"; }));
    }
    break;

    case 203:
        lbTemperatureDown.BeginInvoke(new Action() => {
lbTemperatureDown.Text = rx_buffer[1].ToString(); }));
        lbTemperatureDown.BeginInvoke(new Action() => {
lbTemperatureTop.Text = rx_buffer[2].ToString(); }));
        if (rx_buffer[3] == 1)
        {
            lbHeatingStatus.BeginInvoke(new Action() => {
lbHeatingStatus.Text = "Enable"; }));
        }
        else
        {
            lbHeatingStatus.BeginInvoke(new Action() => {
lbHeatingStatus.Text = "Disable"; }));
        }
        break;

    case 204:
        int temp = 0;
        for(int i = 0; i < (int)rx_buffer[1]; i++)
        {
            temp *= 10;
            temp += rx_buffer[2 + i];
        }
    }
}
```

```
lbHumidityGround.BeginInvoke(new Action() => {
lbHumidityGround.Text = temp.ToString(); }));

lbHumidityAir.BeginInvoke(new Action() => {
lbHumidityAir.Text = rx_buffer[2 + (int)rx_buffer[1]].ToString(); }));

if(rx_buffer[3 + (int)rx_buffer[1]] == 1)
{
lbWater.BeginInvoke(new Action() => { lbWater.Text
= "Yes"; }));
}
else
{
lbWater.BeginInvoke(new Action() => { lbWater.Text
= "No"; }));
}
if (rx_buffer[4 + (int)rx_buffer[1]] == 1)
{
lbWatering.BeginInvoke(new Action() => {
lbWatering.Text = "Enable"; }));
}
else
{
lbWatering.BeginInvoke(new Action() => {
lbWatering.Text = "Disable"; }));
}
break;

case 205:
lbDate.BeginInvoke(new Action() => { lbDate.Text =
rx_buffer[4].ToString() + "." + rx_buffer[5].ToString() + "." +
rx_buffer[6].ToString(); }));
```

```

lbTime.BeginInvoke(new Action(() => { lbTime.Text =
rx_buffer[1].ToString() + ":" + rx_buffer[2].ToString() + ":" +
rx_buffer[3].ToString(); }));
break;

default:
break;
}
}
else if (read_enable == 1)
{
rx_buffer[rx_wr_index++] = (byte)data;
if (rx_wr_index == 64) rx_wr_index = 0;

if (++rx_counter == 64)
{
rx_counter = 0;
}
}
}
}

private void tmrChecker_Tick(object sender, EventArgs e)
{
string[] devices = SerialPort.GetPortNames();

// Remove
for (int i = 0; i < cbDevice.Items.Count; i++)
{
bool remove = true;

```

```

foreach (string device in devices)
{
    if (cbDevice.Items[i].ToString() == device)
    {
        remove = false;
    }
}

if (remove)
{
    cbDevice.Items.RemoveAt(i);

    i = 0;
}
}

// Add
foreach (string device in devices)
{
    bool add = true;
    for (int i = 0; i < cbDevice.Items.Count; i++)
    {
        if (device == cbDevice.Items[i].ToString())
        {
            add = false;
        }
    }

    if (add)
    {

```

					IA52.200БАК.005 ПЗ	Лист
						110
Ізм.	Лист	№ докум.	Підпис	Дата		


```

        cbDevice.Items.Add(device);
    }
}

#region Device
private void btConnection_Click(object sender, EventArgs e)
{
    if (btConnection.Text == "Connect")
    {
        if (cbDevice.SelectedIndex >= 0)
        {
            port = new SerialPort(cbDevice.SelectedItem.ToString(),
9600);

            try
            {
                port.Open();
                receiving = new Thread(ReceiveData);
                receiving.Start();
            }
            catch
            {

            }

            btConnection.Text = "Disconnect";
        }
        else
        {
            MessageBox.Show("You should select one of devices!");
        }
    }
}

```

					IA52.200БАК.005 ПЗ	Лист
						111
Ізм.	Лист	№ докум.	Підпис	Дата		

```

    }
else
{
    try
    {
        if (receiving.IsAlive)
        {
            receiving.Abort();
        }
        if (port.IsOpen)
        {
            port.Close();
        }
    }
catch
{

}

btConnection.Text = "Connect";
}
}
#endregion

```

					IA52.200БАК.005 ПЗ	Лист
						112
Ізм.	Лист	№ докум.	Підпис	Дата		